

Aki Martikainen  
TARJOUSLASKURI

Insinöörityö  
Kajaanin ammattikorkeakoulu  
Tekniikan ja liikenteen ala  
Tietotekniikan koulutusohjelma  
Kevät 2005



**Kajaanin  
ammattikorkeakoulu**

## INSINÖÖRITYÖ TIIVISTELMÄ

Osasto	Tekniikka	Koulutusohjelma	Tietotekniikka
Tekijä(t)			
Aki Martikainen			
Työn nimi			
Tarjouslaskuri			
Vaihtoehtoiset ammattiopinnot		Ohjaaja(t)	
Ohjelmistotekniikka		Raili Simanainen Juha Pennanen	
Aika	8.4.2004	Sivumäärä	52 + 5
Tiivistelmä			
<p>Insinööriyön tarkoituksena oli suunnitella ja toteuttaa Graninge Kainuu Oy:lle ohjelmisto, jonka avulla on mahdollista tehdä asiakkaille tarjouksia sähköstä ja tarvittaessa vertailla tarjouksien edullisuutta asiakkaan saamiin muihin tarjouksiin. Vertailussa otetaan huomioon kilpailevien sähköyhtiöiden siirto- ja myyntihintatiedot. Ohjelmiston hintatietoina käytetään energiamarkkinaviraston julkaisemia virallisia hintatietoja. Hintatiedot voidaan päivittää tarvittaessa myös ohjelmassa.</p> <p>Tarjouslaskurilla voidaan tuottaa nopeasti tarjous- ja sopimusdokumentteja. Hintavertailuun tarvittavat tiedot on mahdollista tallentaa Microsoft Access -tietokantaan. Tietokantaan tallennetut tarjoukset voidaan palauttaa Tarjouslaskuriin tarkastelua tai uutta vertailua varten. Tietueiden poistaminen tietokannasta onnistuu ohjelmiston avulla vaivattomasti.</p> <p>Tarjouslaskuri-ohjelma on toteutettu Visual Basic for Applications -ohjelmointikielellä sisäiseksi osaksi Microsoft Excel -sovellusta. Hintavertailun käyttöliittymä on toteutettu sujuvan käytettävyyden vuoksi mahdollisimman yksinkertaiseksi. Vertailu tehdään päälomakkeella, jossa tarvittavat toiminnot valitaan ja suoritetaan. Hakeminen tietokannasta tehdään erilliseltä hakulomakkeelta SQL-ohjelmointikielen avulla.</p> <p>Ohjelmistoa testattiin läpi ohjelmointivaiheen ajan, ja se läpäisi asiakastestauksen ilman ongelmia. Tarjouslaskuri on ollut Graninge Kainuu Oy:n asiakaspalvelun käytössä 15.3.2005 alkaen. Ohjelmisto on toiminut halutulla tavalla. Insinööriyön tuloksena saatiin aikaan ohjelmisto, joka helpottaa ja nopeuttaa asiakaspalvelun toimintoja. Ohjelmistoa käytetään yleensä Graninge Kainuun lähiverkossa, mutta Tarjouslaskuri voidaan irrottaa toimimaan esimerkiksi kannettavalle tietokoneelle vaikkapa messujen ajaksi.</p>			
Luottamuksellinen			
Kyllä			
Ei X			
Hakusanat			
VBA, Office, tietokanta, SQL			
Säilytyspaikka			



**Kajaanin  
ammattikorkeakoulu**

## ABSTRACT THESIS

Faculty	Faculty of Engineering	Degree programme Information Technology
Author(s)		
Aki Martikainen		
Title		
A Program Comparing Energy Prices		
Optional professional studies	Instructor(s) / Supervisor(s)	
Software Engineering	Raili Simanainen Juha Pennanen	
Date 8 April 2005	Total number of pages 52 + 5	
<p>Abstract</p> <p>The purpose of this Bachelor's thesis was to design and develop a program that compares energy prices between two electric companies. The thesis was made for Grange Kainuu Oy. The program considers the electricity transmission fees and electricity sales prices of the companies. Official electricity prices will be taken from the websites of Energiamarkkivirasto (the Energy Market Authority) to the program. These prices can easily be updated into the program, too.</p> <p>With the program, quotation and contract documents can be created. It is possible to store information in the Microsoft Access database. The information can be returned to the main program for further examination from the database. Data records can be deleted from the database without trouble.</p> <p>The program was developed with the Visual Basic for Applications programming language as an extension for Microsoft Excel. Its user interface was created as simple as possible to make the program fast and easy to use. Searching from the database is made with SQL.</p> <p>The program was tested during its development, and it passed customer testing without problems. The energy price comparing program was introduced on 15 March 2005 and it has worked very well. This thesis resulted in a program which makes working easier and faster for the customer service of Grange Kainuu Oy. The program is mostly used in Grange's local area network, but it also works without LAN and the database, for example at a trade fair.</p>		
Confidential		
Yes		
No            X		
Keywords		
VBA, Office, database, SQL		
Deposited at		

## ALKUSANAT

Insinööritöön tilaajana toimi Graninge Kainuu Oy. Insinööritöön valvojina toimivat Juha Pennanen ja Raili Simanainen. Insinööritöön tekeminen oli erittäin opettavainen ja haastava projekti. Työn tekeminen vei paljon aikaa ja koetteli varmasti monen henkilön kärsivällisyyttä sekä hermoja.

Tahtoisin kiittää suuresti työn tilannutta Graninge Kainuuta erittäin mielenkiintoisen ja mielekkään tehtävän tarjoamisesta. Työ ei olisi valmistunut ilman valvojien Juha Pennanen ja Raili Simanaisen näkemyksiä sekä neuvoja. Perheelleni ja ystäville kuuluu myös suuri kiitos kannustuksesta ja tukemisesta. Monet muut henkilöt ovat olleet antamassa mielipiteitään sekä neuvojaan työstä ja toteuttamassa Tarjouslaskurin testausta. Kiitos siitä.

Kajaanissa 8.4.2005

---

Aki Martikainen

## SISÄLLYS

1 JOHDANTO	6
2 TARJOUSLASKURIN TAVOITTEET JA MÄÄRITTELY	7
2.1 Vaatimukset tehtävää työtä varten	8
2.2 Käyttötapauskaavio ja käyttötapauslomakkeet	10
2.3 Toimintokaavio	13
3 OHJELMISTOKEHITYKSEN VAIHEISTUS	16
3.1 Vaihejakomallit	16
3.2 Kehityksen vaiheet	17
3.3 Työvälineet	21
3.4 Menetelmät	29
4 TOTEUTUS	32
4.1 Suunnittelu	32
4.2 Ohjelmointi ja testaus	35
4.3 Asiakastestaus ja käyttöönotto	46
4.4 Jälkihoito	49
4.5 Saavutetut tulokset	49
5 YHTEENVETO	51
LÄHDELUETTELO	52
LIITTEET	

## 1 JOHDANTO

Insinööriyön tilaaja oli Kajaanissa kotipaikkaansa pitävä Graninge Kainuu Oy, joka on energia-alan palveluyritys. Yhtiön kannalta eräitä tärkeitä vuosia olivat 1995 ja 1998. Vuonna 1995 sähkömarkkinat vapautuivat yritysasiakkaille, ja vuonna 1998 sähkömarkkinat avautuivat myös yksityisasiakkaille.

Sähkölasku koostuu sähkönsiirto- ja sähkömyyntihinnasta sekä verosta. Sähköenergian myyntihinta on asiakkaan kilpailutettavissa sitä myyvien yhtiöiden kesken. Sähkönsiirtopalvelu on ostettava kuitenkin aina siltä verkonhaltijalta, jonka verkkoon asiakas on liittynyt. [1.]

Muuttuneet olosuhteet loivat paineita yhtiölle ottaa käyttöön ohjelma, jonka avulla asiakkaiden pyytämät hintavertailut pystyttäisiin helposti laskemaan. Vuonna 2000 Graninge Kainuulle tehtiin ohjelma, jonka avulla hintavertailujen suorittaminen onnistui. Ohjelma on toiminut luotettavasti jo vuosien ajan, mutta on hiljalleen käynyt nykypäivän vaatimuksille täysin riittämättömäksi. Ohjelma on epälooginen käyttää, tietojen tallentaminen on vaivalloista ja ylläpidettävyyys on hankalaa. Ohjelman avulla tuotettavat tulosteet ovat vaikeasti hallittavia, ja laskureitakin on kertynyt useita eri versiota.

Insinööriyön tavoitteena oli suunnitella ja toteuttaa yhtiölle täysin uusi sähköntarjoushintojen vertailulaskuri, joka korvaa vanhan ohjelman. Tässä dokumentissa käsitellään ohjelmistolle asetetut vaatimukset ja tavoitteet, ohjelmiston kehityksen vaiheet sekä ohjelmiston toteutukseen käytettyjä työvälineitä ja teorioita.

## 2 TARJOUSLASKURIN TAVOITTEET JA MÄÄRITTELY

Insinööriyönä luotavan uuden vertailulaskurin haluttiin olevan helposti hallittava ja nopeasti toimiva sähköntarjoushintojen vertailuohjelmisto, joka helpottaa ja nopeuttaa Graninge Kainuun työntekoa. Ohjelmisto vertailee asiakkaan nykyisen siirto- ja myyntiyhtiön hintoja Graninge Kainuun tarjoamiin sähkönhintoihin. Ohjelman laskemat vertailutiedot tulee näkyä ohjelman päälomakkeella, mutta vertailun tarkempiin tietoihin on tarvittaessa päästävä tutustumaan.

Ohjelman pitää pystyä tallentamaan asiakas- ja vertailutiedot tietokantaan. Tiedot on pystyttävä hakemaan tietokannasta tarvittaessa takaisin tarkasteltavaksi. Tietojä on myös pystyttävä poistamaan tietokannasta ohjelmiston avulla. Lisäksi ohjelmalla olisi voitava tuottaa erilaisia tarjouksia ja sopimuksia.

Tarjouslaskuri ohjelmoidaan Visual Basic for Applications -ohjelmointikielellä Microsoft Excel -taulukkolaskentaohjelman sisäiseksi osaksi. Ohjelmaa käytetään graafisesta käyttöliittymästä. Ohjelman tietokantana toimii Microsoft Access, ja tulosteiden tulostamiseen käytetään Microsoft Wordia, johon halutut dokumentit avautuvat hintavertailun tiedoilla täytettynä ja valmiiksi muotoiltuina.

Tarjouslaskurin käyttäjä on Graninge Kainuun asiakaspalvelun henkilökunta. Ohjelmistoa tullaan käyttämään asiakaspalvelutilanteissa, joten käytettävyyteen ja luotettavuuteen tulee kiinnittää huomiota. Tarjouslaskurin pääasiallinen käyttöpaikka on Graninge Kainuun lähiverkko, mutta tarvittaessa Tarjouslaskuri on voitava asentaa toimimaan kannettavalla tietokoneella esimerkiksi messujen ajaksi. Minimilaitteistovaatimuksina Tarjouslaskurin toiminnalle on asetettu tietokone, jossa toimii Windows XP -käyttöjärjestelmä ja Microsoft Office XP -toimisto-ohjelmisto. Erikseen asennettavien lisäkomponenttien käyttöä tulee välttää.

## 2.1 Vaatimukset tehtävää työtä varten

Tarjouslaskuriin on voitava syöttää asiakas-, käyttöpaikka- ja kulutustiedot. Asiakastietoja ovat mm. nimi, osoite ja puhelinnumero. Käyttöpaikkatietoja ovat mm. käyttöpaikan osoite ja käyttöpaikan numero. Kulutustietoja ovat kokonaiskulutus ja osakulutukset prosenttiosuuksineen. Kokonaiskulutus on voitava jakaa osakulutuksiin, ja se on voitava muodostaa osakulutuksista. Oikea tariffi valitaan kokonaiskulutuksen perusteella automaattisesti. Sulakekoko valitaan kulutustiedoissa. Edellä mainitut tiedot on myös pystyttävä tyhjentämään helposti.

Asiakkaan nykyinen siirto- ja myyntiyhtiö sekä niiden tarjoamat tariffivaihtoehdot noudetaan käyttöliittymän päälomakkeeseen erillisestä taulukosta. Siirto- ja myyntiyhtiö ovat vapaasti valittavissa päälomakkeelta. Oletusmyyntiyhtiö valitaan automaattisesti siirtoyhtiön mukaan. Mikäli siirtoyhtiön alueella on useampi myyntiyhtiö, näytetään kaikki nämä yhtiöt erillisessä taulukossa, josta oikea myyntiyhtiö valitaan. Siirtoyhtiön valintaa helpottamaan on toteutettava linkki-painike, joka avaa erillisen tiedoston, josta tarvittava apu löytyy. Siirto- ja myyntitariffi valitaan edellä valittujen yhtiöiden tuoteryhmistä. Myyntitariffi hintoineen on oltava mahdollista syöttää myös käsin, jolloin voidaan käyttää julkisesta hinnastosta poikkeavia hintoja.

Graninge Kainuun tarjoamat tuoteryhmät haetaan erillisestä ylläpitohinnastosta. Käyttöliittymästä valitaan haluttu tuoteryhmä, ja vain tuoteryhmää vastaavat tariffivaihtoehdot näytetään. Valittuja hintoja on mahdollista muokata käsin päälomakkeelta. Sopimuksen aloitus- ja lopetuspäivämäärän asettamisen ohjelma hoitaa automaattisesti. Aloituspäivämääräksi määräytyy kolme viikkoa eteenpäin vertailupäivämäärästä ja lopetuspäivämääräksi kaksi vuotta ja kyseisen kuukauden viimeinen päivä vertailupäivämäärästä. Päivämäärät on voitava muuttaa tarvittaessa käsin.

Ohjelman on tutkittava erillistä huomautuslistaa, jossa ylläpidetään erilaisia huomautettavia tietoja myynti- ja siirtoyhtiöstä sekä niiden tuotteista. Kun listalla oleva yhtiö valitaan, näytetään käyttöliittymän päälomakkeella huomautus poikkeavasta menettelystä. Huomautus ilmestyy päälomakkeelle, kun sille on tarvetta.



Vertailu suoritetaan päälomakkeen painikkeesta edellyttäen, että kulutus, asiakkaan nykyinen siirto-, myyntiyhtiö ja niiden tariffit sekä Graningen tarjoustuote tarifeineen on valittu. Tiedot siirretään Excelissä olevaan ylläpitotaulukkoon. Vertailu lasketaan Excelissä näiden tietojen avulla. Päälomakkeelle palautetaan nähtäväksi asiakkaan kannalta olennaisimmat tiedot. Näitä ovat asiakkaan nykyisen yhtiön hinta, Graningen tarjouksen hinta sekä hintojen erotus. Vertailun tarkempiin tietoihin on tarvittaessa päästävä tutustumaan. Oletusarvoisesti vertailu tehdään arvonlisäveron kanssa, mutta hinnat on pystyttävä muuntamaan myös arvonlisäverottomiksi.

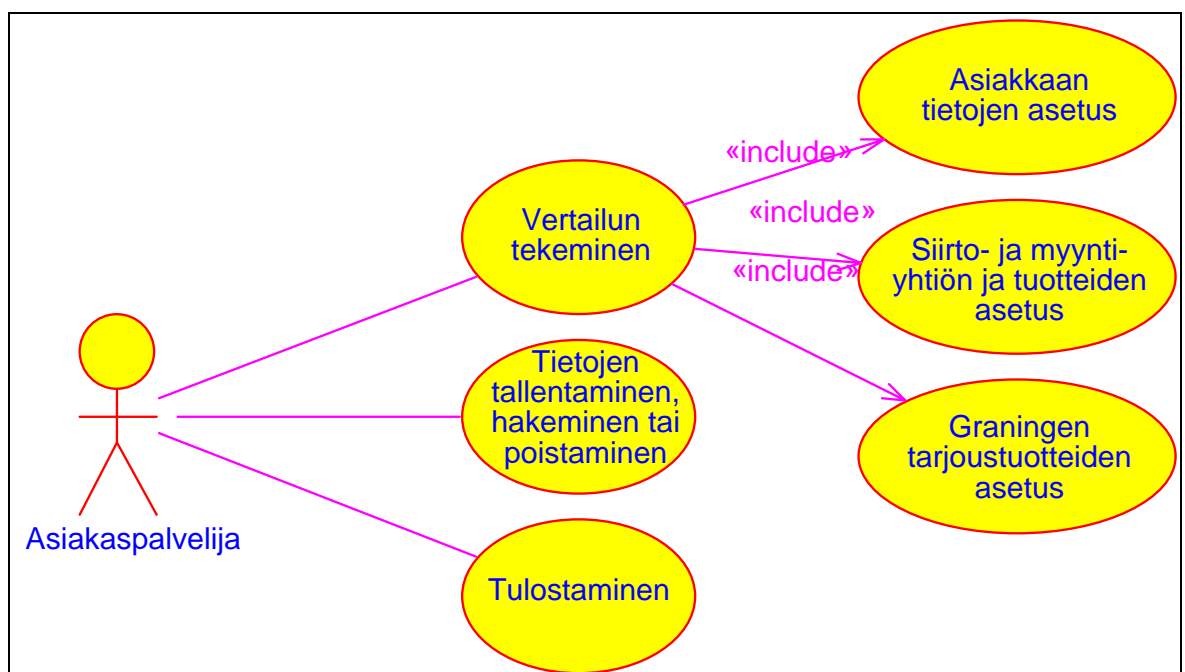
Asiakas- ja vertailutiedot on pystyttävä tallentamaan tietokantaan ja tarvittaessa myös erillisenä tiedostona. Tietokanta tullaan sijoittamaan Graninge Kainuun lähiverkkoon, joten sen on pystyttävä selviämään usean yhtäaikaisen käyttäjän toimiessa tietokannan kanssa. Tiedot tallennetaan tietokantaan päälomakkeella olevan painikkeen avulla. Vertailun tallennus tekee aina uuden tietueen tietokantaan, jotta historiatiedot pysyvät nähtävillä. Tallennus ei ole mahdollista, ellei vertailua ole suoritettu ja vertailun tekijän nimeä syötetty.

Tiedot haetaan tietokannasta omalta lomakkeeltaan. Hakulomake aukaistaan käyttöliittymän päälomakkeelta. Tietokannasta on pystyttävä tekemään hakuja erilaisilla hakuperusteilla. Näitä hakuperusteita ovat mm. nimi, osoite ja vertailupäivämäärä. Hakuehtojen perusteella löydetty tiedot näytetään erillisessä listassa, josta tiedot voidaan siirtää takaisin Tarjouslaskuriin. Tietokannasta on pystyttävä poistamaan haluttaessa tietoja. Tiedot poistetaan samalta lomakkeelta kuin tietojen hakeminen.

Tulostettavat perusdokumentit on voitava valita päälomakkeella olevasta valintalistasta. Perusdokumentteja ovat hintavertailu, sähkönmyyntisopimus ja saate. Jokaisella Graningen tarjoustuoteryhmällä voi olla muista tuoteryhmistä riippumattomia erikoisdokumentteja, jotka ohjelman on pystyttävä näyttämään valintalistassa. Tarjouslaskurin on hoidettava oikean dokumentin avaus ja tietojen siirtäminen ohjelmasta niihin.

## 2.2 Käyttötapauskaavio ja käyttötapauslomakkeet

Ohjelman toiminnallisuus esitetään käyttötapauskaavion eli use case -kaavion avulla. Kuvassa 1 on esitetty Tarjouslaskuri-ohjelman käyttötapaukset. Toimijana kaaviossa toimii asiakaspalvelija, joka kommunikoi järjestelmän kanssa. Kuvan 1 kaaviossa vertailun tekeminen, tietojen tallentaminen, hakeminen tai poistaminen sekä tulostaminen ovat ohjelman yleisiä käyttötapauksia. Vertailun tekeminen ei onnistu ilman välttämättömien tietojen asettamista. Tätä on kuvattu suhteella esimerkiksi vertailun tekemisen ja asiakastietojen asettamisen välillä.



Kuva 1. Tarjouslaskuri-ohjelman käyttötapauskaavio.

Sanallinen kuvaus kuvaa vuorovaikutuksen käyttäjän ja eri käyttötapausten välillä. Vertailun tekemistä kuvaavan käyttötapausten tarkempi sanallinen kuvaus on esitetty taulukossa 1. Tietojen tallentamista, hakemista tai poistamista kuvaavat käyttötapaukset on esitetty taulukossa 2, ja tulostamista kuvaava käyttötapaus on esitetty taulukossa 3.

Taulukko 1. Sanallinen kuvaus ”Vertailun tekeminen” -käyttötapauksesta.

<b>Käyttötapaus:</b>	Vertailun tekeminen
<b>Tavoite:</b>	Vertailu tehdään syöttämällä asiakkaan tiedot, siirto- ja myyntiyhtiön tariffit laskuriin. Näitä tietoja verrataan Graningen tarffiin. Ohjelma vertailee yhtiöiden hinnat. Vertailutietojen perusteella asiakas tietää, onko Graninge Kainuu halvempi kuin asiakkaan nykyinen yhtiö.
<b>Käyttäjät:</b>	Graninge Kainuun asiakaspalvelun henkilö
<b>Esiehdot:</b>	Tarjouslaskuri on avattu ja päälomake on latautunut ruudulle.
<b>Kuvaus:</b>	<p>1. Käyttäjä syöttää asiakkaan asiakas-, käyttöpaikka- ja kulutus-tiedot käyttöliittymän tekstikenttiin ja painaa käyttöliittymän ”Tarkista kulutustiedot” -painiketta, jolloin ohjelma tarkistaa kulutus-tiedot ja valitsee oikean tarffin käytettäväksi.</p> <p>2. Lomakkeelle latautuvat siirto- ja myyntiyhtiöt sekä Graninge Kainuun tarjoamat tarjoustuotteet. Käyttäjä valitsee asiakkaan nykyisen siirto- ja myyntiyhtiön. Yhtiöiden valinnan jälkeen on valittava asiakkaan nykyiset tariffit. Myyntitariffin kuukausi-maksuhintaa ja energiahintoja voi tarvittaessa korjata käyttö-liittymästä.</p> <p>3. Käyttäjä valitsee asiakkaalle sopivan tarffin Graningen tuote-ryhmästä. Myös Graningen hintoja voi tarvittaessa korjata. Käyttäjä painaa ”Suorita vertailu” -painiketta.</p> <p>4. Ohjelma siirtää asiakas-, käyttöpaikka- ja kulutustiedot Excel-työkirjan välilehdelle, jossa suoritetaan hintavertailun tarvittavat laskutoimitukset. Käyttöliittymään tulee näkyville asiakkaan nykyisen yhtiön hinta, Graningen tarjouksen hinta ja niiden erotus. Käyttäjä kuittaa tehneensä vertailun valitsemalla nimensä lomakkeen ”Tämän tarjouksen teki” -alasvetovalikosta.</p> <p>5. Tarvittaessa käyttäjä voi tarkastella vertailua tarkemmin Excelissä painamalla ”Katso vertailua” -painiketta. Excelistä palataan käyttöliittymään painamalla ”Palaa laskuriin” -painiketta.</p>
<b>Poikkeus:</b>	<p>Pakollisia tietoja (kulutustiedot, asiakkaan nykyisten yhtiöiden tiedot ja Graningen tiedot) ei ole syötetty, jolloin ruudulle ilmoitus asiasta. Käyttäjä syöttää/valitsee puuttuvat tiedot.</p> <p>Käyttäjän virhe, eli käyttäjä on valinnut vahingossa väärän yhtiön tai tuotteen, jolloin vertailu menee väärin. Käyttäjä valitsee oikeat tiedot käyttöliittymästä ja suorittaa vertailun uudelleen.</p>
<b>Jälkiehdot:</b>	Tarjouslaskuri on laskenut vertailun ja näyttää vertailun tiedot käyttöliittymässä ja Excelin välilehdellä.

*Taulukko 2. Sanallinen kuvaus ”Tietojen tallentaminen, hakeminen ja poistaminen” -käyttötapauksesta.*

<b>Käyttötapaus:</b>	Tietojen tallentaminen, hakeminen ja poistaminen
<b>Tavoite:</b>	Käyttäjä voi tallentaa tietokantaan tietoja, hakea sieltä tietoja, sekä poistaa tietokannasta tarpeettomia tietoja.
<b>Käyttäjät:</b>	Graninge Kainuun asiakaspalvelun henkilö
<b>Esiehdot:</b>	<p>Tietojen tallennus: Vertailu on suoritettu onnistuneesti ja tarjouksen tekijän nimi on valittuna.</p> <p>Tietojen hakeminen: Käyttäjä on siirtynyt käyttöliittymän päälomakkeelta hakulomakkeelle. Tietokantaan on tallennettu tietoja.</p> <p>Tietojen poistaminen: Käyttäjä on siirtynyt käyttöliittymän päälomakkeelta hakulomakkeelle. Tietokantaan on tallennettu tietoja ja tietoja on haettu lomakkeelle.</p>
<b>Kuvaus:</b>	<p>1. Tietojen tallennus: Käyttäjä painaa ”Tallenna tiedot” -painiketta, jolloin asiakas-, käyttöpaikka ja vertailutiedot tallentuvat tietokantaan.</p> <p>2. Tietojen hakeminen: Käyttäjä syöttää hakuehdot lomakkeelle ja painaa ”Hae tietokannasta” -painiketta. Kaikki hakuehtoja vastaavat tiedot löytyvät tietokannasta. Käyttäjä valitsee oikean vertailun ja painaa ”Vie tiedot laskuriin” -painiketta.</p> <p>3. Tietojen poistaminen: Käyttäjä valitsee poistettavan tiedon listalta ja painaa ”Poista tiedot” -painiketta.</p>
<b>Poikkeukset:</b>	<p>Tietokantaan tallennus, kannasta hakeminen tai poistaminen ei onnistu. Ruudulle tulee ilmoitus viasta. Ohjelmistoa on korjattava.</p> <p>Hakutiedoilla ei löydy haettavia tietoja. Käyttäjä tarkistaa hakuehdot ja suorittaa haun uudelleen.</p>
<b>Jälkiehdot:</b>	Tietojen tallennus, hakeminen ja poistaminen onnistuvat. Ruudulle tulee ilmoitus tapahtuman onnistumisesta. Käyttäjä voi jatkaa ohjelman käyttämistä.

Taulukko 3. Sanallinen kuvaus ”Tulostaminen” -käyttötapauksesta.

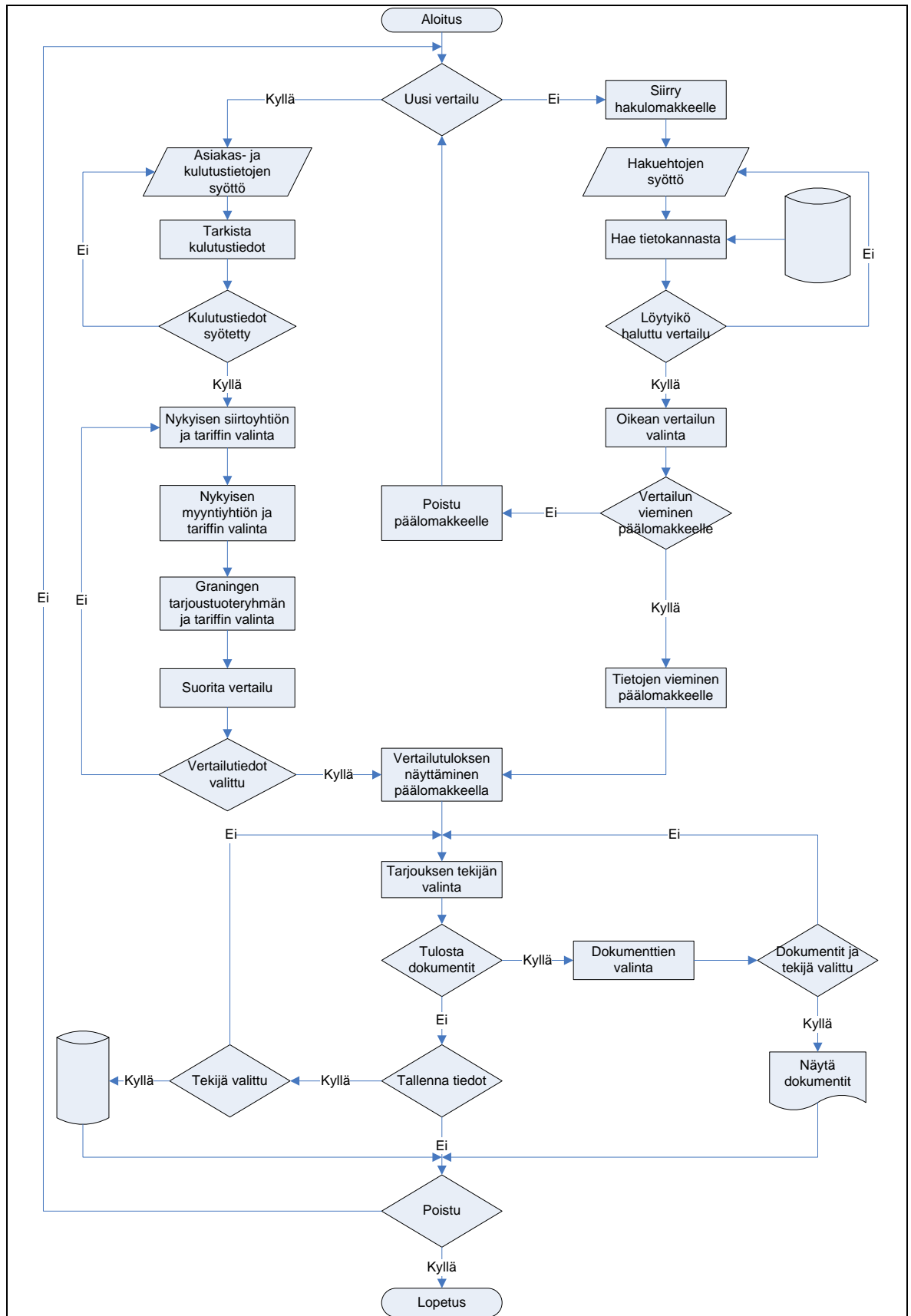
<b>Käyttötapaus:</b>	Tulostaminen
<b>Tavoite:</b>	Käyttäjä valitsee päälomakkeelta haluamansa dokumentit, jotka hän haluaa tulostaa. Tulosteet aukeavat valmiiksi muotoiltuina ja tarpeellisilla tiedoilla täytettynä. Käyttäjä tulostaa dokumentit Wordissa ja jatkaa työskentelyään.
<b>Käyttäjät:</b>	Graninge Kainuun asiakaspalvelun henkilö
<b>Esiehdot:</b>	Vertailu on suoritettu onnistuneesti ja tarjouksen tekijän nimi on valittuna.
<b>Kuvaus:</b>	<p>1. Käyttäjä valitsee päälomakkeelta tulosteet, jotka hän haluaa tulostaa. Käyttäjä painaa ”Tulosta valitut” -painiketta. Microsoft Word -tekstinkäsittelyohjelma aukeaa automaattisesti ruudulle. Wordiin avautuvat tarvittavat dokumentit, jotka käyttäjä on valinnut tulostettavaksi. Dokumentteihin ovat siirtyneet kaikki tarvittavat tiedot. Näitä tietoja ovat mm. asiakas-, käyttöpaikka, kulutus-tiedot sekä tarvittavat vertailutiedot.</p> <p>2. Käyttäjä voi halutessaan vapaasti muotoilla vielä dokumentteja. Käyttäjä tulostaa dokumentit ja sulkee Wordin.</p>
<b>Poikkeukset:</b>	<p>Word ei avaudu tai ohjelmien kommunikoinnissa voi myös olla ongelmia, jolloin tiedot eivät siirry.</p> <p>Väärät tiedot siirtyvät tulostettavaksi. Siirto on suoritettava uudelleen. Mikäli virhe aiheutuu uudelleen, on ohjelmistoa korjattava.</p> <p>Wordiin avautuu eri dokumentti kuin käyttöliittymästä on valittu. Ohjelmistoa tai dokumentin polkutietoa on korjattava.</p>
<b>Jälkiehdot:</b>	Käyttäjän tarvitsemat dokumentit tulostuvat tulostimesta halutun näköisenä ja tarpeellisilla tiedoilla täytettynä. Käyttäjä voi jatkaa Tarjouslaskurin käyttämistä.

### 2.3 Toimintokaavio

Ohjelman tapahtumaketju esitetään toimintokaavion avulla. Ohjelman käyttäminen aloitetaan päälomakkeelta, josta voidaan edetä aloittamalla uuden vertailun tekeminen tai hakemalla vertailu tietokannasta siirtymällä hakulomakkeelle. Uuden vertailun tekeminen aloitetaan syöttämällä asiakas- ja kulutustiedot niille varattuihin kenttiin. Käyttäjä suorittaa kulutustietojen tarkistamisen, jolloin ohjelma tuo valittavaksi siirto- ja myyntiyhtiön sekä Graningen tarjoustuoteryhmän valinnan. Käyttäjä valitsee oikeat yhtiöt ja niiden tariffit ja suorittaa vertailun. Päälomakkeelle ilmestyy Graningen tarjoushinta vuodessa, asiakkaan nykyisen sähkönkulutuksen hinta ja niiden erotus.

Käyttäjä voi tutustua tarkemmin vertailun yksityiskohtiin siirtymällä Excel-työkirjan välilehdelle, jossa tarkat tiedot vertailusta löytyvät. Yksityiskohdista palataan takaisin Tarjouslaskurin käyttöliittymään. Käyttäjä valitsee vertailun tekijän ja tulostaa tarvittaessa haluamansa asiakirjat. Näitä asiakirjoja voivat olla esimerkiksi sähkön hintavertailu, sopimus tai saate. Tarvittaessa vertailu viedään tietokantaan. Sähkön hintavertailu on suoritettu ja käyttäjä voi aloittaa uuden vertailun, hakea tietoja tietokannasta tai poistua ohjelmasta.

Vertailut haetaan tietokannasta hakulomakkeelta. Käyttäjä siirtyy hakulomakkeelle, syöttää haluamansa hakuehdot ja suorittaa haun tietokannasta. Mikäli haluttu vertailu löytyy annetuilla hakuehdoilla, käyttäjä valitsee oikean tietueen ja siirtää tiedot päälomakkeelle. Päälomakkeella vertailua voidaan tarvittaessa muokata, tutustua vertailun yksityiskohtiin tarkemmin tai tulostaa halutut asiakirjat. Tietojen tallentaminen tietokantaan onnistuu myös tarvittaessa. Tarjouslaskuri-ohjelman tärkeimmät toiminnot on esitetty kuvan 2 esittämässä toimintokaaviossa.



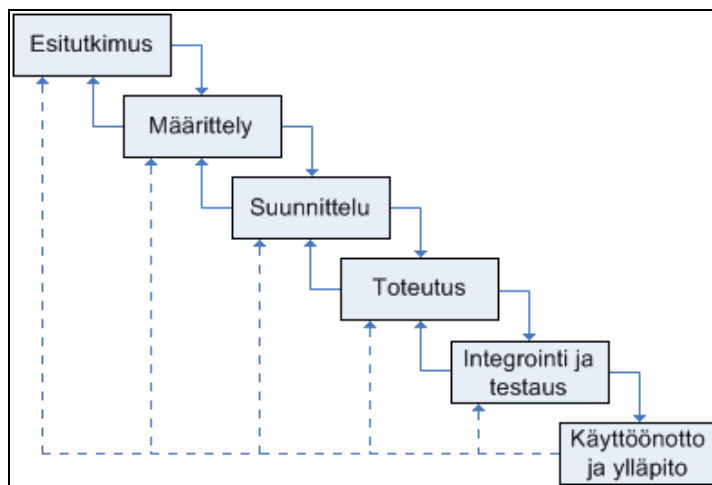
Kuva 2. Tarjouslaskurin tärkeimmät toiminnot esitettyinä toimintokaaviossa.

### 3 OHJELMISTOKEHITYKSEN VAIHEISTUS

Lukujen 3.1 ja 3.2 tiedot pohjautuvat Ilkka Haikalan ja Jukka Märijärven kirjassa Ohjelmistotuotanto esitettyihin tietoihin [2].

#### 3.1 Vaihejakomallit

Ohjelmiston elinkaari on aika, joka alkaa ohjelmiston kehittämisestä jatkuen aina ohjelmiston poistamiseen käytöstä. Ohjelmiston kehitys tai sen koko elinkaari voidaan jakaa erilaisiin vaiheisiin, ns. vaihejakomalleihin. Yleisin vaihejakomalli on vesiputousmalli. Tästä mallista on kehitetty monia erilaisia muunnelmia, mutta niistä jokaisesta on löydettävissä ainakin määrittely-, suunnittelu-, toteutus- ja testausvaiheet. Määrittelyvaihetta edeltää usein esitutkimusvaihe, ja testauksen jälkeen suoritetaan käyttöönotto ja ohjelmiston ylläpito. Eräs vesiputousmallin versio on esitetty kuvassa 3.



Kuva 3. Eräs versio vesiputousmallista.

Kaikkiin vesiputousmallin vaiheisiin liittyy erilaisia laadunvarmistustoimenpiteitä, joita ovat esimerkiksi tarkastukset, katselmukset ja testaus. Tarkastuksilla ja testaamisella tähdätään siihen, että virheet ja väärin toimivat osat saadaan poistettua ohjelmasta jo ennen ohjelman valmistumista. Katselmuksia järjestetään tyypillisesti eri vaiheiden loppumisen yhteydessä. Katselmuksien avulla pyritään varmistamaan, että kaikki vaiheeseen liittyvät tavoitteet on saavutettu ja tulos on riittävän



laadukas. Lisäksi ohjelmistokehitykseen liittyvät koko projektin ajan dokumentointi, laadunvarmistus ja tuotteenhallinta. [2, s. 36-37.]

Todellisessa elämässä ei voida kuitenkaan noudattaa täydellisesti vesiputousmallin mukaista ohjelmiston kehitystä. Syy tähän on, että osa vaatimuksista selviää vasta projektin aikana ja vaatimukset muuttuvat projektin edetessä. Vesiputousmallin mukaisesti voi kuitenkin pyrkiä toimimaan mahdollisimman paljon.

Evo-malli, erilaiset protomallit ja spiraalimalli ovat myös varsin yleisesti käytettyjä elinkaarimalleja. Edellä mainittuja malleja voidaan kutsua myös inkrementaalisiksi malleiksi, koska lopputuotetta kehitetään pieninä inkrementteinä yhden projektin sisällä. Lopullinen ohjelmisto voi olla vaatimusten mukainen esimerkiksi kolmannella inkrementtikierroksella. Jokaisen inkrementtikierroksen lopputulos on kuitenkin jo toimiva järjestelmä, josta puuttuu vielä järjestelmältä vaadittuja osia.

Evo-malli on sikäli vesiputousmallin kaltainen, että se muodostuu sarjasta peräkkäisiä vesiputouksia. Aluksi rakennetaan ohjelmisto, jossa on toimiva ydinjärjestelmä. Jokaisen valmiin vesiputousmallin jälkeen ohjelmiston ominaisuudet ovat lisääntyneet edelliseen versioon nähden. Spiraalimalli koostuu myös toistuvista kierroksista, joissa samat välivaiheet toistetaan jokaisella kierroksella. Spiraalimallissa riskien hallinta otetaan erityisen voimakkaasti huomioon, koska se esiintyy joka kierroksella omana vaiheenaan. Prototyypisessä lähestymisessä tarkoitetaan mitä tahansa työskentelymallia, jossa jotakin kehitettävän tuotteen ominaisuutta kokeillaan käytännössä ennen varsinaisen tuotteen rakentamista. Prototyyppinen lähestyminen sopii hyvin, kun ollaan tekemässä täysin uuden teknisen ratkaisun vaatimaa ohjelmaa. [2, s. 41-45.]

### 3.2 Kehityksen vaiheet

Ohjelmistoa tehtäessä käydään läpi erilaisia kehitysvaiheita. Esitutkimusvaiheessa selvitetään syy ja tarve, miksi ohjelmisto pitäisi toteuttaa tai mitä ohjelmiston pitäisi tehdä. Asiakas asettaa vaatimukset tarvittavalle ohjelmistolle. Asiakasvaatimukset eivät ota millään tavoin kantaa siihen, millä ja miten ohjelmisto tehdään. Käytän-

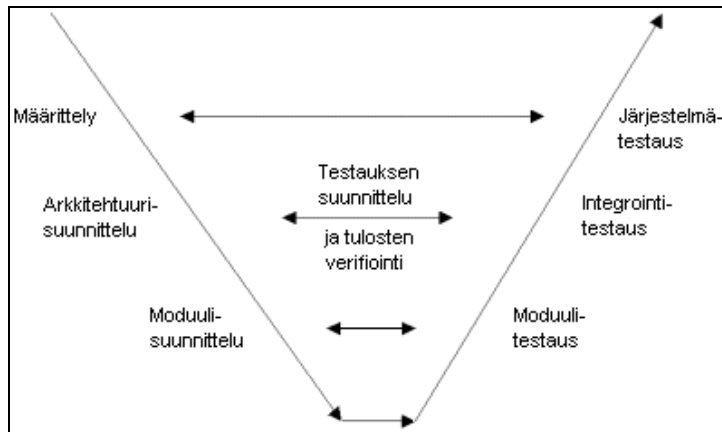
nössä asiakastarpeiden analysointi ja tarkentaminen jatkuu läpi koko määrittelyvaiheen, joten esitukimusta voidaan pitää osana määrittelyvaihetta.

Määrittelyvaiheessa muunnetaan asiakasvaatimukset täsmällisiksi järjestelmävaatimuksiksi. Määrittelyvaiheessa syntyy toiminnallinen määrittely, johon on johdettu luotavan ohjelman ohjelmistovaatimukset asiakkaan antamien vaatimusten määrittelyiden perusteella. Toiminnallisessa määrittelyssä kuvataan ohjelmiston toiminnot, ei-toiminnalliset vaatimukset sekä rajoitukset. Toimintoja ovat mm. ohjelmistolla toteutettavat ominaisuudet, käyttöliittymä ja kommunikointi muiden järjestelmien kanssa. Ei-toiminnallisia vaatimuksia ovat esimerkiksi käytettävyys, toimintavarmuus ja ylläpidettävyys. Rajoituksia ovat käytettävä ohjelmointikieli, minimi laitteisto ja käyttöympäristö.

Suunnitteluvaiheessa suunnitellaan, miten määritellyt toiminnot ja vaatimukset toteutetaan rajoitukset huomioon ottaen. Vaihe voidaan tarvittaessa jakaa kahteen tasoon. Ensin suunnitellaan ohjelmiston arkkitehtuuri, jossa ohjelma jaetaan moduuleihin ja määritellään moduulien väliset rajapinnat. Arkkitehtuurikerroksia ovat mm. käyttöliittymä-, sovelluslogiikka- ja tietokantakerros. Toiseksi suunnitellaan ohjelmasta erotettavissa olevien kokonaisuuksien, moduulien, sisäinen rakenne. Arkkitehtuurisuunnittelusta syntyvää dokumenttia kutsutaan tekniseksi määrittelyksi, jossa kuvataan mm. toteutuksen ratkaisuperiaatteet, tietokanta-arkkitehtuuri ja moduulien tehtävät ja rajapinnat.

Toteutusvaiheessa suunnitelmat toteutetaan valituilla tekniikoilla. Ohjelma ohjelmoidaan ensimmäiseen virheettömään käännökseen asti.

Testausvaiheessa on tarkoitus löytää ohjelmistosta virheitä. Työvaiheita ovat testauksen suunnittelu, ympäristön luonti, testien suorittaminen sekä tulosten tarkastelu. Usein testaukseen ja korjauksiin kuluu yli puolet projektin resursseista. Testauksessa käytetään usein V-mallia, joka koostuu moduuli-, integrointi- ja järjestelmätestauksesta. [2, s. 38-41.] Kuvassa 4 on esitelty testauksen V-malli.



*Kuva 4. Testauksessa yleisesti käytetty V-malli.*

Moduulitestauksessa ohjelmaa testataan moduuleittain etsien vikoja yksittäisistä moduuleista. Moduulit eivät välttämättä ole itsenäisesti toimivia, joten niiden testaamiseen tarvitaan testimoduuleita edustamaan yksikköä kutsuvaa ohjelmaa ja korvaamaan niitä moduuleita, joita testattava moduuli kutsuu. Yksittäisen moduulin toimintaa verrataan moduulin suunnitteluun ja tekniseen määrittelydokumenttiin. Testauksen suorittaa yleensä moduulin toteuttaja.

Integrointitestauksessa valmiita moduuleja kootaan yhteen ja testataan moduulien yhteistoimintaa, painopiste on moduulien välisten rajapintojen testauksessa. Testaus etenee usein kokoavasti, eli alimman tason moduuleista ylöspäin. Integroinnin tuloksia verrataan tekniseen määrittelyyn.

Tekijä suorittaa ohjelmalle vielä järjestelmätestauksen, jossa järjestelmä käydään läpi suunnitelmallisesti kokonaisuutena. Tuloksia verrataan määrittelydokumenttiin ja tarkastetaan ohjelman täyttävän asiakkaan asettamat vaatimukset. Asiakas-testauksessa asiakas testaa ohjelman omassa käyttöympäristössään. Testauksen aikana löytyneet korjataan ohjelmoijan toimesta. Virheen korjaaminen voi aiheuttaa ohjelman muissa osissa virheitä, joten ohjelmalle täytyy suorittaa järjestelmä-testaus uudelleen. Tätä vaihetta kutsutaan regressiotestaukseksi. Testauksen läpi käynyt ja hyväksymiskriteerit täyttänyt ohjelma voidaan ottaa käyttöön. [2, s. 286-289.]

Lasilaatikko- ja mustalaatikkotestaus ovat yleisesti käytettyjä testausmenetelmiä, joita tässä insinööriyössäkin on käytetty. Lasilaatikkotestaus (white-box) perustuu ohjelman kontrollirakenteen testaamiseen. Testitapaukset luodaan tarkastelemalla ja analysoimalla ohjelman lähdekoodia. Testitapauksissa pyritään suorittamaan ja testaamaan koodi mahdollisimman kattavasti. Lausekattavuudessa testit valitaan siten, että ohjelman jokainen koodirivi suoritetaan vähintään kerran. Lausekattavuuteen voi olla hyvin vaikeaa päästä, ja yleensä testauksessa jäädyäänkin alle 90 prosentin kattavuuteen. Päätöskattavuudessa edellytetään, että ehtolauseet saavat testien aikana molemmat arvonsa. Ehtokattavuudessa päätöksen kaikkien ehtojen on saatava kaikki eri arvonsa. Päätös/ehtokattavuudessa testitapauksissa on suoritettava sekä päätös, että ehtokattavuus. Polkutestauksessa pyritään kulkemaan kaikki ohjelman läpi kulkevat suorituspolut. Käytännössä tämä on mahdollista vain yksittäisten funktioiden tapauksessa, jolloin eri polkuvaihtoehtoja on vielä testauksen kannalta järkevä määrä. Lasilaatikkotestausta käytetään tyypillisesti moduuli- ja integrointitestauksessa. [2, s. 289-293.]

Mustalaatikkotestaus (black-box) on toiminnallinen testaus, jossa ei tunneta ohjelman kontrollirakennetta tai siitä ei välitetä. Testaus keskittyy ohjelman ulkoisesti havaittavan toiminnallisuuden ja piirteiden testaamiseen. Testitapaukset valitaan jakamalla syöteavaruus ekvivalenssiluokkiin, joissa jokaisessa on vain yksi saman luokan edustaja. Lisäksi mukaan voidaan ottaa luokkien rajoilla olevia syötteitä. Syötteet annetaan ohjelmalle ja näiden syötteiden tuloksia verrataan tiedossa oleviin tietoihin, kuinka ohjelman olisi pitänyt toimia. Näin testaus voidaan rajata resurssien mukaiseksi, koska täydellinen testaus on syötteiden valtavan määrän johdosta käytännössä mahdotonta toteuttaa. Mustalaatikkotestauksen avulla ohjelmasta voidaan löytää puuttuvat toiminnallisuudet ja virheelliset suorituspolut, joita lasilaatikkomenetelmillä ei voida todeta. [2, s. 289-290.]

### 3.3 Työvälineet

#### Visual Basic for Applications

VBA on tulkittu, proseduraalinen ohjelmointikieli. Tulkittu ohjelmointikieli tarkoittaa sitä, että ohjelmoijan kirjoittama lähdekoodi, jonka VBA-tulkki ensin lukee Basic-kielellä, muunnetaan VBA-tulkin avulla Windowsin suorituskelpoiseksi binäärikoodiksi. Tietokone pystyy käsittelemään tämän binäärikoodin ja saa aikaan ohjelmoijan haluaman lopputuloksen. [3.]

VBA-kielellä on lähes sama kapasiteetti kuin normaalilla Visual Basic -kielellä muutamia poikkeuksia lukuun ottamatta. VBA:lla luotuja ohjelmia ei voi kääntää itsenäisesti toimiviksi tiedostoiksi. VBA:lla luodut ohjelmat eivät käytä omaa muistiavaruutta, kuten normaalilla Visual Basicilla tehdyt ohjelmat tekevät. VBA on osa sovellusta, johon se on integroitu, esimerkiksi Microsoft Exceliä. VBA:n avulla voidaan luoda itsenäisiä sovelluksia, jotka tarvitsevat isäntsovellustaan vain toimiakseen. VBA:n avulla voi kuitenkin myös muokata ja laajentaa itse isäntäohjelmaa tai sen toimintoja sopimaan juuri omiin tarpeisiinsa. [4.]

Mikä tahansa ohjelmointikieli (jossa on Automation-tuki), joka voi viestiä objekti-kirjaston kanssa, pystyy ohjaamaan Microsoft Office -sovelluksia. VBA keskustelee esimerkiksi Excelin objekti-kirjaston kanssa. VBA:n avulla pystytään itse asiassa ohjaamaan mitä tahansa objekti-kirjaston omaavaa sovellusta. Office-sovellusten lisäksi monissa muissa Microsoftin ja muiden valmistajien sovelluksissa on objekti-kirjasto. VBA:ssa ja Visual Basicin itsenäisessä 6.0-versiossa on sama kieliohjelma sekä lähes samanlaiset editorit. [5.]

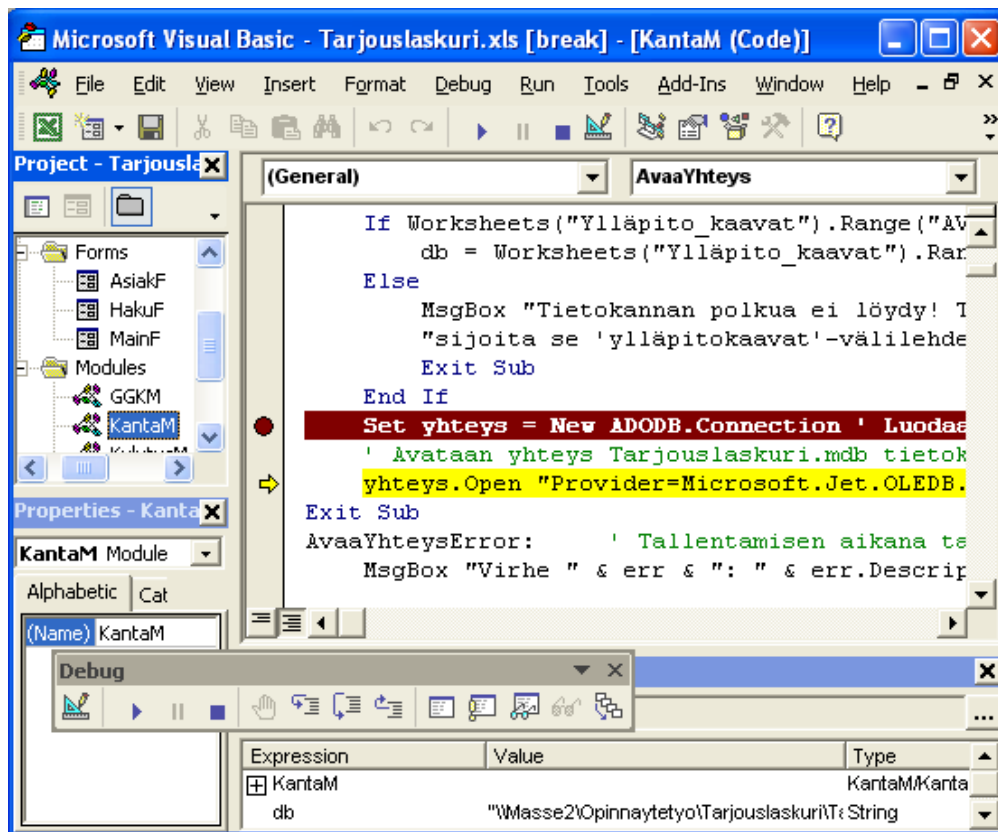
#### Microsoft Office

Microsoft Office on ohjelmistopaketti, johon on koottu yhteen useita Microsoftin tekemiä toimistosovelluksia. Office-paketista on olemassa useita erilaisia versioita, mutta yrityksissä tyypillisesti käytetään Professional-versiota, joka sisältää muun muassa Microsoft Word-, Excel- ja Access-ohjelmat. Edellä mainittuja Office-paketin ohjelmia hyödynnettiin tässä insinöörityössä.

Access on tietokantaohjelma tietojen hallintaan ja säilömiseen. Tietokanta voi sisältää joukon taulukoita, lomakkeita, kyselyitä, raportteja ja moduuleita. Näitä tietokantaobjekteja käytetään hyväksi tietojen hallitsemisessa ja esittämisessä. Accessissa tiedot talletetaan taulukkoihin. Taulukoita on mahdollista yhdistää relaatioiden avulla, jolloin saadaan aikaan relaatiotietokanta. Taulukoiden, lomakkeiden ja raporttien luomiseen voidaan käyttää ohjattua toimintoa, joka tekee Access-tietokantaohjelman käyttämisen helpoksi kokemattomallekin käyttäjälle. Tietokantaa voidaan hallita myös erillisen ohjelman avulla. Tällöin tietokantaa voidaan käyttää avaamatta itse Access-ohjelmaa. [6, s. 833-834.]

### Visual Basic Editori

Visual Basic Editori on kaikkien Office-sovellusten mukana tuleva täydellinen sovelluskehitysympäristö. Visual Basic Editori koostuu erilaisista toimintopaleteista ja -ikkunoista. Editorin avulla voidaan luoda ja suorittaa Office dokumentteihin liittyvää VBA-koodia ja se sisältää toimintoja jotka nopeuttavat koodin kirjoittamista ja suorittamista. Editoriin on luotu virheenkäsittelyominaisuuksia, joita ovat mm. koodin suorittaminen rivi riviltä ja koodiin sijoitettavat pysähtymiskohdat. Virheenkäsittelyn aikana Editorissa voidaan tarkkailla muuttujien arvoja, mikä helpottaa virheen etsimistä. Kuvassa 5 on Visual Basic Editorin sovelluskehitysympäristö.



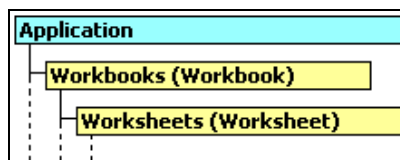
Kuva 5. Visual Basic Editorin sovelluskehitysympäristö.

## Objektikirjastot

Jokaisella Office-sovelluksella on oma objektikirjastonsa. Näissä objektikirjastoissa on oma objektihierarkia ja omat erityiset objektit sekä objektikokoelmat. Objektihierarkialla tarkoitetaan sitä, miten eri objektit liittyvät toisiinsa. Kaikkien Office-sovellusten objektihierarkioiden ylin objekti on Application, esimerkiksi Excel- tai Word-sovellus itse. Loput objektit ovat ylimmän Application-objektin jälkeläisiä. Objekteista on myös muodostettu valmiita objektikokoelmia. Niiden avulla voidaan ohjelmoida suoraan kokonaista objektiryhmää. [7, s. 78.]

## Microsoft Excel -objektikirjasto

Visual Basic sisältää objekteja, jotka ovat Excelin osia. Näistä tutuimpia ovat Application-, Workbook-, Worksheet- ja Range-objekti. Kuvassa 6 on esitetty pieni osa Excelin objektihierarkiaa.



Kuva 6. Microsoft Excel -objektihierarkian ylin objekti on Sovellus.

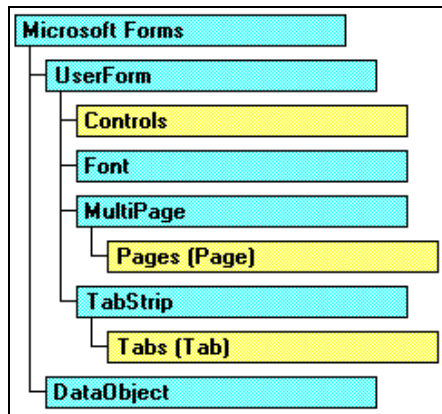
Excelin Application-objektin ominaisuuksilla hallitaan sovelluksen yleistä käyttäytymistä, sovellusikkunaa ja sen avulla päästään käsiksi objektihierarkiassa alempana oleviin objekteihin. Workbook-objektin avulla voidaan hoitaa mm. tiedoston luomiseen, avaamiseen, tallentamiseen ja sulkemiseen liittyviä toimintoja. Range-objekti voi edustaa yhtä solua, solualuetta tai vaikkapa kokonaista riviä. Range-objektilla ei ole kokoelmaobjektia.

## Microsoft Forms -objektikirjasto

Microsoft Forms -kirjaston objektien avulla voidaan luoda lomakkeita kehitettävään sovellukseen. Lomakkeita käytetään viestintään käyttäjän kanssa ja tietojen syöttämiseen sekä tulostamiseen. Lomakkeiden avulla luodaan siis ohjelman käyttöliittymä, joten niiden käyttäminen parantaa käyttäjäystävällisyyttä.

Lomakkeet rakentuvat Office-sovelluksissa samalla tavalla. UserForms-kokoelmaan sisältyy UserForm-objekti, joka muodostaa jokaisen sovelluksessa käytettävän lomakkeen. Kaikki muistiin ladatut lomakkeet kuuluvat kokoelmaan. Lomakkeelle sijoitetaan tarvittavat ohjaimet (esimerkiksi painikkeita ja tekstiruutuja), joiden avulla voidaan lähettää sopiva ohje lomakemoduulille suoritettavaksi. [7, s. 95-97.] Kuvassa 7 on esitetty Microsoft Forms -objektihierarkia.

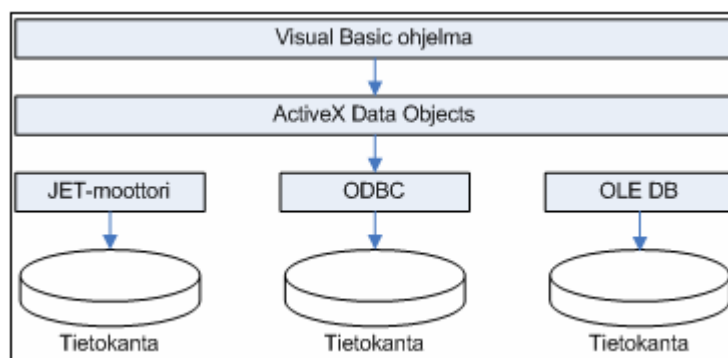




Kuva 7. Microsoft Forms -objektihierarkia.

### ADO-objektikirjasto

ActiveX Data Objektit toimivat liittymänä Visual Basicin ja tietokannan välillä. ADO-objektikirjastossa on objekteja, ominaisuuksia ja metodeja, joita voidaan käyttää tiedon saantiin ja käsittelyyn. ADO:n tietotähteitä ovat muun muassa Microsoft Access-, Microsoft SQL Server-, Oracle RDBMS-tietokannat. ADO:n avulla ODBC ja OLE DB -ohjaimet voidaan liittää tietokantaan ilman uudelleen ohjelmointia. ADO-kirjastoa voidaan käyttää esimerkiksi VB-, VBA-, ja C++ -sovelluskehitysympäristöissä, joten sen avulla voidaan rakentaa laiteriippumattomia ja tehokkaita sovelluksia. Kuvassa 8 on esitetty Visual Basic ohjelman kommunikointi ADO:n avulla tietokantaan. [8, s. 213-225.]



Kuva 8. ADO:n toimiminen liittymänä Visual Basicin ja tietokannan välillä.

ADO-objektimalli ei ole hierarkkinen. Se koostuu seitsemästä objektista, joista Connection, Command ja Recordset ovat pääobjekteja. Connection-objekti tarjoaa yhteyden tietolähteeseen ja pitää yhteyttä avoimena. Command-objekti sisältää SQL tai muun käskyn, joka suoritetaan tietolähteessä. Sitä käytetään usein tallennetun koodin suorittamiseen täydessä tietokantapalvelimessa. Recordset-objektin avulla voidaan käsitellä tietokanta kyselyn tietueita, esimerkiksi selata tai luoda uusia tietueita. Tietuejoukko voidaan hakea tietokannasta esimerkiksi SQL-lauseen avulla. [8, s. 217-222.]

### OLE Automation -objektikirjasto

Automation-objekti (ennen OLE Automation) on COM-standardiin perustuva tekniikka, jonka avulla voidaan hyödyntää sovelluksen toimintoja toisessa sovelluksessa. Automation mahdollistaa ohjelmien välisen kommunikoinnin, datan vaihtamisen ja toistensa kontrolloimisen. Automationia tukevat sovellukset pystyvät paljastamaan toimintojaan ja ominaisuuksiaan objektikokoelmina Automationin kautta. [9.]

Kaikki Microsoft Officeen kuuluvat sovellukset osaavat hyödyntää Automation-objekteja. Näiden objektien avulla voidaan esimerkiksi avata tai luoda uusi Word-dokumentti Excelistä, muokata sitä, tulostaa ja tallentaa se nimellä sekä sulkea dokumentti lopuksi.

### Tietokannat

Tietokanta on tietokokonaisuus, johon on kerätty loogisesti yhteenkuuluvia tietoja tiettyä tarkoitusta varten. Tallennettu tieto voidaan hakea tarkasteltavaksi tietokannasta, muokata sitä ja poistaa tarpeeton tieto myöhemmin. Tietokannassa olevia tietoja voidaan hallinnoida ja muokata esimerkiksi SQL-tietokantakielen avulla. Tietoja hallinnoidaan tietokannan hallintajärjestelmä ohjelmiston avulla. Tällaisia ohjelmistoja ovat muun muassa Access, SQL Server ja Oracle. Erityyppisiä tietokantoja on olemassa useita. [10, s. 4.]

Yleisesti käytettyjä tietokantatyyppejä ovat kortistotietokanta, hierarkkinen tietokanta, verkkotietokanta ja relaatiotietokanta. Kortistotietokanta on suhteellisen pienissä tietomäärissä yleisesti käytetty perustietokanta, jossa kaikki toisiinsa liittyvät tiedot sijoitetaan yhteen taulukkoon. Tällaisen tietokannan perustaminen on nopeaa ja ei vaadi suurta suunnittelua. Kortistotietokanta on suurien tietomäärien tallentamisessa tehotonta, tilaa vievää ja raskasta, sillä kaikki yhteiset tiedot on toistettava jokaisessa uudessa tietueessa. [6, s. 835.]

Hierarkkiset tietokannat olivat ensimmäisiä tietokantajärjestelmiä ja niiden sisältämät tiedot järjestettiin hierarkkiseen muotoon. Rakenne on kuitenkin jäykkä ja turhilta tiedoilta ei pystytä välttymään. Verkkotietokannassa tietojen väliset suhteet pyrittiin kuvaamaan joukkoina hierarkioiden sijasta. Verkkotietokannan ja hierarkkisen tietokannan erottaa toisistaan verkkotietokannan mahdollisuudesta antaa lapsitauluilleen useampi kuin yksi äititaulu. Tällaisen tietokannan toteuttaminen ja ylläpitäminen on hankalaa. [11, s. 3.]

Nykyisin yleisin käytössä oleva tietokantatyyppi on relaatiotietokanta. Relaatiotietokannat ovat pääsääntöisesti SQL-pohjaisia. Relaatiotietokannan ajatus on, että tietoja ei syötetä useaan kertaan, vaan käytetään useita taulukoita, jotka yhdistetään toisiinsa relaatioiden avulla. Relaatiotietokanta on helppokäyttöinen ja joustava, mutta se vaatii tietokoneelta myös eniten suoritustehoa. [10, s. 5.]

Relaatiotietokanta koostuu 2-ulotteisista tauluista. Taulut koostuvat riveistä ja sarakkeista. Yhtä riviä tietokannassa kutsutaan tietueeksi. Tietueet koostuvat kentistä ja sisältävät tietokannan tiedot. Sarakkeet ovat taulukon ominaisuuksia. Taulukoiden sisältämät tietueet yksilöidään perusavaimen avulla, jotta kukin tieto on vain kertaalleen tietokannan tauluissa. Perusavain on määritettävä jokaiseen tietokannan tauluun. Tietokannan taulujen väliset yhteydet hoidetaan viiteavaimen avulla. Yhteystyyppejä taulujen välillä ovat yhden suhde yhteen, yhden suhde moneen ja monen suhde moneen. [11, s. 5-8.]

Relaatiotietokannan tietojen on oltava ristiriidattomia ja oikeellisia toimiakseen oikein, tietokantaan on asetettu tiettyjä rajoitteita. Näitä ovat avain- ja viite-eheys. Avaineheys estää perusavaimen olevan tyhjä ja huolehtii, että usealla tietueella ei

voi olla sama perusavaimen arvo. Viite-eheys yhteydessä varmistaa, että perusavaimen ja viiteavaimen arvot ovat samoja. Viite-eheys estää myös tiedon virheelisen muokkaamisen tai poistamisen tietokannasta. [10, s. 11-12.]

## SQL

Structured Query Language on erityisesti tietokantojen käsittelyä varten luotu tietokantakieli. SQL-kieltä voidaan käyttää tietokannan rakenteen määrittämiseen ja muuttamiseen, erilaisten kyselyiden tekemiseen. Kyselyt ovat SQL:n vahvin osa-alue ja kieltä käytetään yleisimmin niiden tekemiseen. Tietojen lisääminen, muokkaaminen, poistaminen, tietokannan valtuuksien ja turvallisuuden hoitaminen on myös tehtävissä SQL:n avulla. Ohjelmoijat tarvitsevat tapahtumankäsittelyn ohjaamista, upotettua SQL:ää ja kohdistimia luodessaan tietokantapohjaisia ohjelmia. [11, s. 38-39.]

SQL-kielessä tietokannan sisällön käsittelyä kutsutaan Data Manipulation Language -osaksi. DML-osaan kuuluvat tietojen hakemiseen, muuttamiseen, lisäämiseen ja poistamiseen liittyvät käskyt. Taulukossa 4 on esitetty muutamia yleisiä DML-osaan kuuluvia SQL-käskyjä. [11, s. 40.]

*Taulukko 4. Tärkeitä SQL-kielen DML-osaan liittyviä käskyjä.*

Komento	Merkitys
SELECT	Suoritetaan kysely lause, jolla haetaan tietoa tietokannan tauluista. Käskyssä on aina oltava SELECT- ja FROM-määre.
DELETE	Poistetaan tietueita tietokannan taulukosta. Käskyssä on aina oltava SELECT- ja FROM-määre.
DISTINCT	Poistetaan kyselystä mahdolliset toistuvat rivit
WHERE	Määritetään mitkä rivit otetaan mukaan hakutuloksiin
LIKE	Mahdollistetaan haut vain osittain täsmäävillä arvoilla
BETWEEN	Voidaan etsiä tiettyjen arvojen välissä olevia arvoja
ORDER BY	Järjestellään halutun haun tulokset tiettyyn järjestykseen
INSERT INTO	Lisätään tietueita tietokannan taulukkoon.
UPDATE	Muutetaan tietueen tietoja tietokannan taulukossa

SQL-kielessä tietokannan rakenteen määrittelyyn liittyviä komentoja kutsutaan Data Definition Language -osaksi. DDL-osaan kuuluvat muun muassa tietokannan

taulujen määrittelyyn ja luomiseen liittyvät käskyt. Taulukossa 5 on esitetty muutamia yleisiä DDL-osaan kuuluvia SQL-käskyjä. [11, s. 60].

*Taulukko 5. Yleisiä SQL-kielen DDL-osaan liittyviä käskyjä.*

Komento	Merkitys
CREATE TABLE	Luodaan uusi taulu tietokantaan
ALTER TABLE	Muutetaan olemassa olevan taulukon rakennetta
DROP TABLE	Poistetaan taulu tietokannasta
CREATE INDEX	Luodaan taulukkaan sarakkeelle indeksi, joka nopeuttaa hakemista tietokannasta
DROP INDEX	Poistetaan taulukon sarakkeeseen luotu indeksi

### 3.4 Menetelmät

Tarjouslaskuri-ohjelman kehitysvaiheet etenivät pitkälti vesiputousmallia mukaillen. Asiakkaalta saatiin ehdotus kehitettävästä ohjelmistosta ja hyvin tarkat vaatimukset siihen. Näiden vaatimusten pohjalta tehtiin vaatimustenmäärittely-dokumentti. Asiakasvaatimusten ja vaatimustenmäärittelydokumentin perusteella luotiin tekninen suunnitelma. Tekniseen suunnitelmaan johdettiin ohjelmistovaatimukset, jotka määrittelevät tehtävän järjestelmän. Tekninen suunnitelma toimi perustana ohjelmiston toteutuksen aikana. Ohjelmistoa testattiin koko kehityksen ajan ja virheitä. Testauksen tuloksena löydettyjä virheellisiä tai väärin toimivia moduuleja korjattiin sitä mukaa kuin niitä havaittiin. Toteutusvaiheen valmistuttua ohjelma vietiin asiakastestaukseen. Asiakastestauksen jälkeen ohjelmistoon tehtiin asiakastestauksen aikana ilmenneiden virheiden tarvitsemat korjaukset. Ohjelmistolle suoritettiin korjausten jälkeen regressiotestaus, jotta voitiin varmistua ohjelmiston toimivuudesta. Regressiotestauksen jälkeen Tarjouslaskuri otettiin virallisesti käyttöön.

## UML

Unified Modeling Language on standardoitu oliosuuntautunut mallinnuskieli. UML on ohjelmistojen mallien kuvaustapa ja graafinen suunnittelukieli, joka määrittelee kielen rakenteen ja merkityksen. UML:ää voidaan käyttää monien erityyppisten järjestelmien mallintamiseen ja järjestelmäkehityksen eri vaiheissa vaatimusten määrittelystä lopullisen ohjelman testaukseen. [12, s. 2-7.]

UML-mallinnuskieleen kuuluvia osia ovat näkymät, mallinnuselementit, kaaviot ja yleiset merkinnät. Erityyppisten näkymien avulla voidaan kuvata kehitettävää järjestelmää. Jokainen näkymä vastaa yhtä järjestelmän projektiota ja kuvaa yhtä piirrettä siitä. Kaavioissa mallinnuselementit on järjestetty kuvaamaan tiettyä järjestelmän piirrettä. Kehitettävästä järjestelmästä luodut mallit sisältävät yleensä useita kaikentyyppisiä kaavioita. Mallinnuselementit ovat asioita ja ideoita, joita kaavioissa esitetään. Mallinnuselementit kertovat yksiselitteisesti mallinnuselementin merkityksen. Mallinnuselementillä on sitä vastaava näkymäelementti kaaviossa. UML:n kaavioissa voidaan käyttää yleisiä merkintöjä. Näillä merkinnöillä voidaan antaa lisätietoa, joita ei voida mallinnuselementtien avulla antaa. [12, s. 11-24.]

Tässä työssä UML:ää on käytetty käyttötapauskaavion ja toimintokaavion tekoon. Käyttötapauskaaviossa esitellään toimijat ja niiden yhteydet järjestelmän tarjoamiin käyttötapauksiin. Käyttötapaus on järjestelmässä olevan yhden toiminnon kuvaus. Käyttötapaukset on purettu sanalliseksi kuvaukseksi käyttötapauslomakkeisiin. Toimintokaaviossa näytetään Tarjouslaskurin tärkeimpien tapahtumien kulku aikajärjestyksessä.

## Hyvä ohjelmointitapa

Insinööriyönä ohjelmoitu Tarjouslaskuri-ohjelma on pyritty tekemään hyvien ohjelmointitapojen mukaan. Tämä tarkoittaa käytännössä muun muassa sitä, että lähdekoodin proseduurit, funktiot ja muuttujat on pyritty nimeämään kuvaavasti. Lähdekoodi on sisennetty ohjelman rakenteen mukaan yhtenäisesti läpi kaikkien lähdekoodin. Lähdekoodin proseduurit ja funktiot, jotka liittyvät samaan ohjelmiston osaan on sijoitettu samoihin moduuleihin, eli modulaariseksi rakenteeksi. Proseduureista ja funktioista on pyritty tekemään mahdollisuuksien mukaan yleiskäyttöisiä. Lähdekoodin jokaisen moduulin, proseduurin ja funktion tehtävä tai tarkoitus on kommentoitu koodiin lyhyesti. Lisäksi koodista löytyy yksittäisiä kommentteja, jotka antavat lisätietoa kyseisestä kohdasta.

## 4 TOTEUTUS

Insinööriyön tekeminen aloitettiin kuukauden tutustumisjaksolla Visual Basic- ja Visual Basic for Applications -ohjelmointikieleen sekä tietokantoihin. Tämän tutustumisvaiheen aikana Visual Basic ja Visual Basic Editori alkoivat käydä tutuiksi. Insinööriyön tilannut Graninge Kainuu Oy antoi informaatiota ja koulutusta sähkönmyyntiin liittyvistä asioista ja hintavertailujen tekemisestä.

Asiakas toimitti hyvin tarkat tavoitteet ja vaatimukset, millainen ohjelmiston tulisi olla. Asiakkaan vaatimuksista luotiin vaatimustenmäärittelydokumentti. Dokumenttiin kirjattiin yleiskuvaus asiakkaasta, kehitettävästä tuotteesta sekä tuotteen käyttäjästä ja käyttöympäristöstä. Dokumenttiin kuvattiin asiakkaan ohjelmalle antamat vaatimukset ja tavoitteet. Kaikki toiminnalliset ja ei-toiminnalliset ominaisuudet kuvattiin yksityiskohtaisesti ja lisäksi otettiin huomioon rajoitteet, joita järjestelmässä on. Lisäksi dokumenttiin liitettiin insinööriyön alustava aikataulu.

### 4.1 Suunnittelu

Vaatimustenmäärittelydokumentin avulla tehtiin ohjelmistosuunnitelma, jossa suunniteltiin muun muassa toteutettava ohjelmisto toimintoihin ja ominaisuuksiin. Dokumentissa käsitellään ohjelman mallinnusta (käyttötapauskaavio ja toimintokaavio), ohjelmistoarkkitehtuuria, käytettäviä toteutusratkaisuita ja tietokantaa sekä ohjelman ajoympäristöä. Ohjelmistosuunnitelma käytettiin perustana työlle koko kehityksen ajan. Seuraavaksi on esitetty ohjelmistosuunnitelman tärkeimpiä kohtia.

#### Arkkitehtuuri

Ohjelmiston arkkitehtuurilla tarkoitetaan ohjelmiston koostumista erilaisista rakenneosista ja näiden rakenneosien suhteista. Arkkitehtuurin tarkoituksena on antaa kehittäjälle malli siitä, miten uusi ominaisuus toteutetaan järjestelmään. Arkkitehtuurirakenteita ovat esim. kerrokset, moduulit ja alijärjestelmät. [2, s. 314-315.]



Tarjouslaskurissa käytetään monitasoarkkitehtuuria, johon kuuluu käyttöliittymä-, sovelluslogiikka- ja tietokantakerros. Tarjouslaskurin yksinkertainen arkkitehtuurikuvaus on esitelty kuvassa 9.



*Kuva 9. Tarjouslaskurin yksinkertainen arkkitehtuurikuvaus.*

Käyttöliittymäkerroksen tehtävä on välittää tietoa, se ei ota osaa tiedon käsittelemiseen. Käyttöliittymä välittää käyttäjän antamat komennot ja toiminnot sovelluslogiikkakerrokselle. Sovelluslogiikka on kerros käyttöliittymän ja tietokannan välissä. Sovelluslogiikkaosa ei ole tiedon näyttämistä tai tallentamista varten, se käsittelee tietoa. Sovelluslogiikkakerroksen moduuleissa olevat proseduurit ja funktiot käsittelevät käyttäjän ja käyttöliittymän vaatimat toimintapyynnot. Tietokantakerros on tiedon tallentamista ja hakemista varten. Tietokantakerros kommunikoi sovelluslogiikan kanssa. Tietokannassa suoritetaan sovelluslogiikan vaatimat toimintapyynnot. Kerrosten välinen kommunikointi on kaksisuuntaista ja tapahtuu rajapintojen avulla.

## Kirjastot

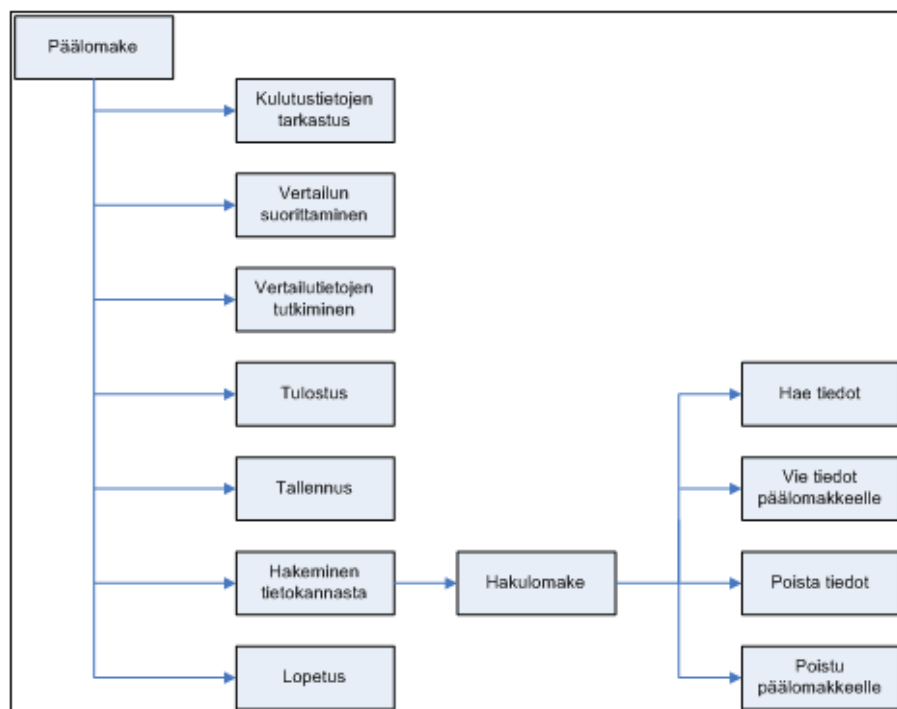
Tarjouslaskuri toteutettiin VBA-kielellä, sisäiseksi osaksi Exceliä, joten mukaan oli otettava muutamia kirjastoja tarvittavien objektien ja ominaisuuksien mukaan saamiseksi. Tarjouslaskuri-ohjelmassa käytettävät kirjastot on esitetty taulukossa 6.

Taulukko 6. Tarjouslaskurissa tarvittavat ohjelmointikirjastot.

Kirjasto
Visual Basic for Applications
OLE Automation
Microsoft Office 10.0 Object Library
Microsoft Excel 10.0 Object Library
Microsoft Word 10.0 Object Library
Microsoft Forms 2.0 Object Library
Microsoft Office 10.0 Object Library
Microsoft ActiveX Data Objects 2.5 Library

### Käyttöliittymä

Tarjouslaskurin käyttöliittymä on se väline, jolla käyttäjä on vuorovaikutuksessa järjestelmän kanssa. Tarjouslaskuria käytetään usein asiakaspalvelutilanteissa, kuten puhelun aikana, joten käyttöliittymän päälomakkeelle suunniteltiin kaikki vertailuun, tulostamiseen ja tallentamiseen liittyvät toiminnot. Tietokannasta tehtäville hauille suunniteltiin oma lomakkeensa. Kuvan 10 kaaviossa on esitetty Tarjouslaskurin looginen käyttöliittymän kartta.



Kuva 10. Tarjouslaskurin looginen käyttöliittymän kartta.

## Tietokanta

Tarjouslaskurin tietokantana käytetään Microsoft Access -tietokantaa. Aluksi suunniteltiin Tarjouslaskurin tietokantatyypiksi relaatiotietokantaa, joka olisi nykyaikainen ratkaisu. Se olisi helppokäyttöinen sekä joustava ylläpidonkannalta. Myös tietokannan koko pysyisi pienenä, koska samaa tietoa ei tarvitse kantaan syöttää kahta kertaa. Tietokanta päätettiin toteuttaa pohdiskelujen jälkeen kuitenkin niin sanottuna kortistotietokantana, eli tietokanta sisältää vain yhden taulun, johon kaikki tiedot tallennetaan. Syynä tähän ratkaisuun oli asiakkaan tarpeet ja mielipide sekä tekijän vähäinen kokemus relaatiotietokannoista. Pelkona oli, että relaatiotietokannan suunnittelu ja toteutus veisivät niin paljon aikaa, että projekti ei pysy aikataulussaan. Asiakkaan mielestä itse hintavertailu ja oikein rakentuvat tulosteet olivat tärkeämmät kuin tietokanta, joten tietokantaan päätettiin insinöörityön aikana panostaa hieman vähemmän. Tietokannan muuttaminen relaatiotietokannaksi myöhemmin on täysin mahdollista. Liitteessä A on esitetty Tarjouslaskuri-ohjelman tietokannan Vertailu-taulu.

### 4.2 Ohjelmointi ja testaus

Tarjouslaskuri-ohjelma ohjelmoitiin pienissä osissa, joista muodostui toimiva kokonaisuus vaatimustenmäärittely- ja ohjelmistodokumentin suunnitelmien mukaan. Ohjelman toiminta kirjoitettiin tapahtuma-aliohjelmiin, proseduureihin ja funktioihin. Käyttöliittymän ohjainten toiminnasta vastaavat tapahtuma-aliohjelmat sijoitettiin lomakemoduuleihin. Tiedon käsittelyn hoitavat proseduurit ja funktiot sijoitettiin pääsääntöisesti erillisiin moduuleihin. Jokaisella erillisellä moduulilla on oma tehtäväkokonaisuus. Esimerkiksi KantaM-moduuli käsittelee ainoastaan tietokantaan liittyviä asioita. Ohjelman toteutus eteni karkeasti jaettuna taulukon 7 esittämissä vaiheissa.

*Taulukko 7. Tarjouslaskurin karkeat ohjelmointivaiheet.*

	Ohjelmointivaihe
1.	Käyttöliittymän toteutus
2.	Kulutustietojen logiikan toteutus
3.	Siirto- ja myyntiyhtiön tuotteiden ja tariffivaihtoehtojen hakeminen päälomakkeelle sekä niiden valitseminen vertailua varten
4.	Graningen tarjoustuotteiden ja tariffivaihtoehtojen hakeminen päälomakkeelle sekä niiden valitseminen vertailua varten
5.	Vertailutietojen siirtäminen Exceliin taulukkoon ja vertailun suorittaminen
6.	Vertailu-, sopimus- ja saatedokumentin valitseminen ja tulostaminen
7.	Tietokantaan tallennus, sieltä hakeminen ja poistaminen

### Käyttöliittymä

Tarjouslaskuri-ohjelman toteuttaminen aloitettiin luomalla käyttöliittymä ohjelmalle. Visual Basic -kielessä käyttöliittymä koostuu lomakemoduuleista. Lomakemoduuli muodostuu lomakkeen ulkoasusta ja toiminnasta. Ulkoasu ja toiminta tehdään omissa ikkunoissaan.

Asiakkaan vaatimuksena käyttöliittymälle oli toteuttaa vertailuun liittyvät tiedot yhdellä lomakkeella ja saada käytettävyys mahdollisimman yksinkertaiseksi. Tarjouslaskurin käyttöliittymä muodostui kahdesta lomakemoduulista, pää- ja hakulomakkeesta. Päälomakkeelle sijoitettiin kaikki tarpeelliset ohjaimet, kuten syöttökentät, painikkeet ja valikot. Käyttöliittymän ohjaimet nimettiin heti alkuvaiheessa mahdollisimman kuvaavasti. Syötettyjen kulutustietojen perusteella päälomakkeelle avautuvat yleis- tai aikatariffin mukaiset valinnat tehtäviksi. Hakulomakkeelle toteutettiin syöttökentät, painikkeet ja lista-kontrolli, joiden avulla voidaan suorittaa erilaisia hakuja tietokannasta ja siirtää halutut tiedot takaisin päälomakkeelle muokkausta varten.

Ennen ohjelmoinnin aloittamista käyttöliittymä katselmoitiin asiakkaan kanssa. Katselmointi tuotti muutamia pieniä muutoksia käyttöliittymään. Vaiheen lopputuloksena saatiin aikaan Tarjouslaskurin käyttöliittymä, joka vastasi asiakkaan antamia vaatimuksia. Kuvassa 11 on esitetty toteutettu käyttöliittymän päälomake.

**Tarjouslaskuri - Versio 1.0**

<b>Asiakastiedot</b> Sukunimi <input type="text"/> Etunimi <input type="text"/> Katuosoite <input type="text"/> Postinumero <input type="text"/> Postitoimipaikka <input type="text"/> Puhelinnumero <input type="text"/> Sähköpostiosoite <input type="text"/>	<b>Käyttöpaikan tiedot</b> Katuosoite <input type="text"/> Postinumero <input type="text"/> Postitoimipaikka <input type="text"/> Käyttöpaikanro <input type="text"/>  <b>Asiakas- ja käyttöpaikkatiedot</b> <input type="button" value="Hae vertailu"/> <input type="button" value="Tyhjennä"/>	<b>Sähkönkäyttötiedot</b> Yhteensä kWh/v <input type="text"/> <input type="button" value="Apulinkki"/> Pöytä/taivarkipv <input type="text"/> % Yö/muu aika <input type="text"/> % <input type="button" value="Tarkista käyttötiedot"/> Sulakekoko <input type="text"/> <input type="button" value="Tyhjennä / vaihda tuote"/>																								
<b>Vertailutiedot: Yleissähkö</b> Siirtoyhtiö <input type="text"/> Myyntiyhtiö <input type="text"/> Siirtotuote <input type="text"/> Myyntituote <input type="text"/> <table border="1"> <thead> <tr> <th>Tuote</th> <th>e/kk</th> <th>snt/kWh</th> <th>Voimassaoloaika</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>		Tuote	e/kk	snt/kWh	Voimassaoloaika																					<b>Myyntiyhtiöt</b> Yhtiön nimi <input type="text"/> Kunta <input type="text"/>  <b>Vertailupainikkeet</b> <input type="button" value="Suorita vertailu"/> <input type="button" value="Katso vertailua"/>
Tuote	e/kk	snt/kWh	Voimassaoloaika																							
<b>GGK vertailutiedot: Yleissähkö</b> Tarjoustuote <input type="text"/> Sopimus alkua <input type="text"/> Sopimus päättyy <input type="text"/> <table border="1"> <thead> <tr> <th>Tuote</th> <th>e/kk</th> <th>snt/kWh</th> <th>Voimassaoloaika</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>		Tuote	e/kk	snt/kWh	Voimassaoloaika																					<b>Hintavertailu</b> e/vuosi <input type="text"/> e/vuosi <input type="text"/> Grange Kainuu <input type="text"/> Nykyinen yhtiö <input type="text"/> Erotus vuodessa <input type="text"/> <b>Valitse tulostettavat asiakirjat</b> <input type="text"/> <b>Tämän tarjouksen teki</b> <input type="text"/> <b>Toimintapainikkeet</b> <input type="button" value="Avaa valitut"/> <input type="button" value="Tallenna"/> <input type="button" value="Poistu laskurista"/>
Tuote	e/kk	snt/kWh	Voimassaoloaika																							

© Grange Kainuu Oy

Kuva 11. Tarjouslaskurin päälomake aloitustilassa.

## Kulutustiedot

Toteutusvaiheessa ohjelmoitiin luvussa 2.1 esitettyjen vaatimusten mukainen logiikka kulutustiedoille (sähkönkäyttötiedoille). Syötettäessä käyttöliittymän kenttiin vain kokonaiskulutus, ohjelma siirtyy yleistariffi-tilaan. Aikatariffi-tilaan päästään syöttämällä kenttiin osakulutustietoja. Ohjelma osaa laskea ja näyttää kulutustiedot minkä tahansa kahden kulutustiedon perusteella. Ohjelma ilmoittaa käyttäjälle mahdollisista virheellisistä syöttötiedoista, jolloin tiedot on tarkastettava.

Kuvassa 12 kulutustietoihin on syötetty kokonaiskulutus (3467) vuodessa ja kalliin sähkön prosenttiosuus (33) kokonaiskulutuksesta. Näiden tietojen avulla ohjelma on laskenut kulutustiedot ja hakenut käytettävissä olevat sulakekoot alasvetovalikkoon.

Sähkönkäyttötiedot	
Yhteensä kWh/v	3467
Pv/talviarkipv	1144
Yö/muu aika	2323
Tarkista käyttötiedot	
Sulakekoko	3*25 A
Tyhjennä / vaihda tuote	

Kuva 12. Kulutustietojen valitseminen päälomakkeelta.

Kulutustietoihin liittyvät funktiot ja proseduurit sijoitettiin omaan moduuliinsa. Moduulin toiminta testattiin ehto- ja mustalaatikkotestauksen avulla. Vaiheen tuloksena saatiin aikaan vaatimusten mukainen ja käyttäjän kannalta yksinkertainen sekä vaivaton toteutus kulutustietojen syöttämiselle.

#### Vertailutietojen valinta

Vaiheen toteutuksena täytyi muun muassa saada kaikki Suomessa toimivat siirto-, myyntiyhtiöt ja Graningen tarjoustuoteryhmä sekä niiden tarjoamat tariffivaihtoehdot valittavaksi päälomakkeelle. Näistä valitaan tarvittavat tiedot vertailun suorittamista varten. Tarkemmin tietojen hakemista koskevia vaatimuksia on kuvattu luvussa 2.1.

Ohjelmointi aloitettiin hakemalla Excelin Ylläpito\_kaavat-välilehdellä ylläpidettävästä taulukosta kaikki Suomessa sähkönsiirtoa tarjoavat yhtiöt päälomakkeella olevaan alasvetovalikkoon. Valitun siirtoyhtiön tariffivaihtoehdot tuodaan energiamarkkinaviraston julkaisemasta taulukosta, joka on liitetty Excelin välilehdeksi. Ohjelma tarkastaa onko valitulle siirtoyhtiölle niin sanottua oletusmyyntiyhtiötä tarjolla. Mikäli oletusmyyntiyhtiö löytyy, tuodaan myyntiyhtiö ja kyseisen yhtiön tariffivaihtoehdot päälomakkeelle valittavaksi samoin tavoin kuin siirtoyhtiön tiedot. Myyntiyhtiö voidaan vaihtaa, mikäli asiakas ei käytä oletusmyyntiyhtiön palveluita. Viimeisenä toteutettiin huomautus-kenttä, johon ohjelma hakee tietoja erillisestä huomautus-taulukosta valitun siirto- ja myyntiyhtiön perusteella. Mikäli valituille yhtiöille ei ole syötetty huomautuksia, huomautus-kenttä jää tyhjäksi. Tariffin

valinta tapahtuu valitsemalla käytetty vaihtoehto listasta. Myyntitariffin hinnat ohjelmoitiin tuotavaksi valinnan jälkeen tekstikenttiin, joten kilpailutettavissa olevia myyntihintoja voidaan tarvittaessa muokata käsin mahdollisimman todellisen vertailulopputuloksen varmistamiseksi.

Graningen tarjoustuoteryhmä ja niiden eri tariffivaihtoehdot haetaan omista taulukoistaan. Ohjelmakoodin toteutus on hyvin samanlainen, kun asiakkaan nykyisten yhtiöiden kanssa ja toteutuksessa voitiin hyödyntää jo luotuja proseduureja ja funktioita. Lisäksi luotiin aliohjelma, joka asettaa sopimuksen aloitus- ja päättymispäivämäärän automaattisesti päälomakkeen tekstikenttiin.

Tämä oli insinööriyön ohjelmoinnin vaikein ja haastavin osuus. Muuttuvia tekijöitä on paljon ja lisäksi ohjelmoinnissa täytyi ottaa huomioon onko kyseessä yleis- vai aikatariffi, koska näissä valittavat tiedot ovat toisistaan poikkeavia ja haetaan eri tietolähteistä. Testauksessa hyödynnettiin pääsääntöisesti lasilaatikkotestausmenetelmiä. Testausta suoritettiin jokaisen valmiin funktion ja proseduurin jälkeen. Valmiit aliohjelmat yhdistettiin ja testattiin niiden keskinäinen yhteensopivuus. Integroititestauksen aikana huomattiin monta kertaa, että aliohjelmien yhdistämisen jälkeen jokin osa alkoi toimia väärin.

Tehokkaaksi tavaksi integroinnista johtuneita virheitä oli etsiä ja korjata debuggaamalla koodia rivi riviltä, muuttujien ja objektien arvoja tarkkaillen. Virheellinen toiminta löytyi usein aiemmin jo toimiviksi todetuista aliohjelmista. Vaiheen aikana huomattiin, että testaaminen tuottaa parhaita tuloksia suunnittelemalla testitapaukset etukäteen. Ohjelmoinnin, valtavan testausmäärän ja virheiden paikallistamisen sekä korjaamisen kautta päästiin vaadittuun lopputulokseen. Vaiheen valmistuttua tulokset katselmoitiin asiakkaan kanssa. Kuvassa 13 on esitetty hintavertailuun tarvittavien valintojen tekeminen päälomakkeelta.

Vertailutiedot: Yö-/vuodenaikasähkö

Siirtoyhtiö: Graniinge Kainuu Oy Myyntiyhtiö: Mäntsälän Sähkö Oy

Siirtotuote: Myyntituote

Tuote	e/kk	snt/kWh	Voimassaoloaika
Aikasähkö	3,00	4,25	1.1-31.12 ma-su 7-22
Kausisähkö	3,00	3,10	1.1-31.12 ma-su 22-7
Aikasähkö, wattiklubi	3,00		
Kausisähkö, wattiklubi	3,00		

Aikasähkö, wattiklubi 3,00 4,25 3,10

Myynti: Mäntsälän Sähkö myy huhtikuussa sähköä erittäin halvalla.

GGK vertailutiedot: Yö-/vuodenaikasähkö

Tarjoustuote: Oma-alue, kaksi vuotta Sopimus alkaa: 24.4.2005 Sopimus päättyy: 30.4.2007

Tuote	e/kk	snt/kWh	Voimassaoloaika
Yösähkö	3,00	4,15	1.1-31.12 ma-pe 7-22, la 15-20
Säästöyösähkö	3,00	3,15	1.1-31.12 ma-pe 22-7, la 7-15, 20-24, su 0-
Vuodenaika	3,00		
Säästövuodenaika	3,00		

Säästöyösähkö 3,00 4,15 3,05

Kuva 13. Vertailutietojen valitseminen päälomakkeelta.

### Vertailun suorittaminen

Ensin ohjelmoitiin tutkiminen, ovatko tarvittavat vertailutiedot valittu päälomakkeelta. Ohjelma huomauttaa käyttäjää viestillä valintojen puuttumisesta. Vertailutiedot siirretään Exceliin, jossa vertailun suorittaminen toteutetaan. Laskutoimituksissa lasketaan syötetyillä kulutustiedoilla vuosittainen sähköstä maksettava hinta. Kokonaishinta muodostuu seuraavasti.

$$\text{Sähkön kokonaishinta} = \text{sähkönmyynti} + \text{sähkönsiirto}$$

Kokonaishinta lasketaan asiakkaan nykyiselle yhtiölle ja Graningen tarjoukselle. Nämä hinnat vähennetään toisistaan, jolloin saadaan hintaero selville. Yhtiöiden kokonaishinnat ja erotus palautetaan käyttöliittymän päälomakkeelle nähtäväksi. Käyttäjä voi poistua käyttöliittymästä Excelin välilehdelle tarkastelemaan hinta-



vertailun tarkempia tietoja. Käyttöliittymään palattaessa kaikki tiedot ovat säilyneet näkyvillä.

Hintavertailu olisi voitu hoitaa myös suoraan ohjelmakoodissa, mutta työn tilaajan vaatimusten mukaan vertailu suoritettiin Excelissä. Vertailu suoritetaan arvonlisäveron kanssa, mutta tarvittaessa laskelmat voidaan muuntaa arvonlisäverottomiksi.

Tarjouslaskurin hintavertailuosion valmistuttua testaus painottui mustalaatikkotestaukseen, jossa ohjelmalle annettiin ennakoon valittuja syötteitä käyttöliittymästä ja tutkittiin saatiinko aikaan haluttu lopputulos. Tarjouslaskurin antamia vertailutuloksia verrattiin Graningen vanhaan laskuriin ja Helsingin Energian nettilaskuriin (<http://www.helsinginenergia.fi>). Testaus tehtiin useilla eri kombinaatioilla sekä yleis- että aikatariffissa. Taulukossa 8 on esitetty erään testitapauksen syötöarvot. Kyseisestä testistä saadut tulokset on esitetty taulukossa 9.

*Taulukko 8. Testitapauksessa käytetyt arvot.*

	Yhtiö	Tuoteryhmä	Energiamaksu	Päiväenergia	Yöenergia
Siirto	Graninge Kainuu Oy	Yösähkö	20,10	2,49	1,55
Myynti	Graninge Kainuu Oy	Yösähkö	3,50	4,95	3,40

*Taulukko 9. Saadut testitulokset.*

Ohjelma	Myynti €/ vuosi		Siirto €/ vuosi		Kokonaishinta
Tarjouslaskuri	296,25	+	425,55	=	694,80
Graningen vanha laskuri	296,25	+	425,55	=	694,80
Helsingin Energian laskuri	269,25	+	425,35	=	694,60

Tarjouslaskurin arvot täsmäsivät tarkalleen Graninge Kainuun vanhan laskurin arvojen kanssa. Graningella vanhan version on todettu vuosien saatossa toimivan oikein. Helsingin Energian laskuri tuotti aina noin 20 senttiä halvemman tuloksen testitapauksesta riippumatta. Tämä ero johtui sähköveron pyöristämisestä, sillä Helsingin Energian laskuri käyttää sähköverona arvoa 0,906 ja Tarjouslaskurissa käytetään veron arvona 0,91. Suoritettuja vertailutuloksia tutkimalla ja analysoimalla voidaan päätellä Tarjouslaskurin hintavertailijan toimivan oikein.

## Tulosteet

Tarjouslaskurin pitää pystyä hallitsemaan erilaisia tulosteita. Käyttäjän haluamat dokumentit valitaan päälomakkeen listasta, jossa näytetään jokaiselle Graningen tarjoustuoteryhmälle valittavissa olevat tulosteet. Tulosteet voidaan avata, mikäli vertailu on suoritettu ja vertailun tekijä valittu. Ohjelma huomauttaa käyttäjää viestillä valintojen puuttumisesta. Ohjelma hakee avattavan tulosteen polun Excelin Ylläpito\_kaavat-välilehdeltä, avaa Word-ohjelman ja halutun asiakirjan polkutiedon perusteella. Wordin avaaminen Tarjouslaskurissa tapahtuu Visual Basic for Applications -kielellä seuraavalla tavalla.

```
Sub AvaaWord(pPolku As String)
    Dim wdApp As Word.Application, wdDoc As Word.Document
    On Error Resume Next
    Set wdApp = GetObject(, "Word.Application")
    If err.Number <> 0 Then ' Onko Word jo avattuna.
        Set wdApp = CreateObject("Word.Application") '
    End If
    On Error GoTo 0 ' Nollataan mahdollinen virhe
    On Error Resume Next
    Set wdDoc = wdApp.Documents.Open(pPolku) ' Asetetaan työkirja
    If err.Number <> 0 Then
        MsgBox "Virhe avattaessa tulostettavaa tiedostoa." _
            & vbCrLf & "Etsi virhe ja korjaa se." & vbCrLf & "Syy: " _
            & err.Number & " " & err.Description, vbExclamation, _
            "Virhe tulostettavaa tiedostoa avattaessa"
        On Error GoTo 0 ' Nollataan virhe
        Exit Sub
    End If
    wdApp.Visible = True ' Avataan Word
    wdDoc.Activate ' Aktivoidaan työkirja
    Set wdDoc = Nothing
    Set wdApp = Nothing
End Sub
```

Valittavina olevat Word-asiakirjat on luotu ja muotoiltu etukäteen. Asiakirjat saavat kaikki tarvittavat asiakas- ja vertailutiedot yhdistämistoiminnon avulla Excelistä Tarjouslaskurin Ylläpito\_sopimukset-välilehdeltä. Kun asiakirja avataan, näkyvät kaikki halutut tiedot asiakirjassa.

Käyttäjä voi tulostaa dokumentin suoraan Wordista ja palata Tarjouslaskuriin. Tapa hoitaa tulosteet Wordin kautta ylitti kaikki odotukset, se on hyvin joustava tapa, kaikki käyttäjät osaavat käyttää Wordiä ja erikoistapauksissa dokumentteja on hyvin helppo muokata tilanteen vaatimaksi. Tulosteiden toimivuuden testaaminen hoidettiin suorittamalla erilaisia vertailuja, avaamalla eri dokumentteja ja tarkastamalla tuliko tulosteisiin oikeat tiedot. Liitteissä B ja C on esitetty ohjelman avulla tuotetut sähkönhintavertailu- ja sähkönmyyntisopimusasiakirja.

## Tietokanta

Tarjouslaskurin avulla tehdyt vertailut täytyi saada talteen tietokantaan. Tiedot on voitava palauttaa tietokannasta takaisin laskuriin tai poistaa tietoja tietokannasta. Tietokantaan tallentaminen toteutettiin suoraan päälomakkeelta. Tallentaminen on mahdollista, mikäli vertailu on suoritettu ja vertailun tekijä valittu. Ohjelma huomauttaa tarvittaessa käyttäjää suorittamaan vaaditut toiminnot ennen tietokantaan tallentamista.

Tietokannasta hakeminen toteutettiin omalta lomakkeeltaan suunnitelmien mukaan. Tietokantayhteys luodaan ja avataan ADO-objektikirjaston Connection-objektin avulla seuraavasti Microsoft Jet -moottoria ja OLE DB -rajapintaa hyödyntäen.

```
Set yhteys = New ADODB.Connection ' Luodaan uusi yhteys
db = "\\Masse2\Tarjouslaskuri\Tarjouslaskuri.mdb"
yhteys.Open "Provider=Microsoft.Jet.OLEDB.4.0; " & "Data Source=" & db & ";"
```

Hakeminen voidaan toteuttaa erilaisten hakuperusteiden avulla. Hakuehdoilla löydettyt tiedot näytetään lomakkeen listassa, josta valitut tiedot voidaan siirtää tietokannasta päälomakkeelle tarkasteltavaksi tai poistaa tietokannasta kokonaan.

Tietokantaa hallitaan ADO-objektikirjaston Recordset-objektin avulla. Usean käyttäjän sattuessa tietokantaan yhtä aikaa, on hoidettu Recordset-objektin open-metodin yhteydessä annettavilla kursori- ja lukitusargumentilla. Tietokannan avaaminen ja hakeminen SQL:n avulla, lisääminen ja poistaminen tietokannasta voidaan suorittaa esimerkiksi seuraavien koodirivien avulla.

```
haku = "SELECT * FROM Vertailu WHERE Sukunimi like '" & Snimi & "%' and Etunimi like '" _
& Enimi & "%' and Osoite like '" & Osoite & "%' and Pnumero like '" & Pnro & "%' and _
Ppaikka like '" & Ppaikka & "%' and GGKtarjous like '" & Tuote & "%' and Vertailupvm like '" _
& Pvm & "%' ORDER BY Sukunimi, Etunimi"
Set tietue = New ADODB.Recordset ' Luodaan uusi recordset-objekti
tietue.Open haku, yhteys, adOpenKeyset, adLockOptimistic ' Avataan yhteys Vertailu tauluun
tietue.addnew ' Lisätään Vertailu-tiluun tietue
tietue.fields("Sukunimi").value = txtSnimi.Text
tietue.update ' Päivitetään muutos
tietue.delete ' Poistetaan kohdalla oleva tietue
```

Tietokantayhteyden varaamat resurssit vapautetaan ja yhteys tietokantaan suljetaan seuraavasti.

```
tietue.Close
Set tietue = Nothing
yhteys.Close
Set yhteys = Nothing
```

Ohjelman Hakulomake on esitetty kuvassa 14.

**Tarjouslaskuri - Vertailun hakeminen tietokannasta**

Haku tietokannasta

Sukunimi:  Etunimi:

Katuosoite:  Postinumero:

Postitoimipaikka:  Tarjoustuote:

Päivämäärä:

Hakuehdoilla löytyi 2 tietuetta.

Nimi	Katuosoite	Postinro	Postitoimip.	Vertailun tekijä	Vertailu pvm
Martikainen Aki "testi"	Testitie 1	87100	Kajaani Aikatariffi	Aki Martikainen	15.03.2005
Martikainen Aki "testi"	Testitie 1	87100	Kajaani Yleistariffi	Aki Martikainen	15.03.2005

Kuva 14. Tarjouslaskurin Hakulomake.

Tietokannan toimivuutta testattiin koko tietokantavaiheen toteutuksen ajan. Lasi-laattikotestauksessa edettiin aliohjelma kerrallaan ja toimivat kokonaisuudet koottiin kokonaisuuksiksi. Kokonaisuuksille suoritettiin integrointitestausta. Mustalaattikotestauksessa tietokantaan syötettiin valittuja ääriarvoja. Tietokannan toimintaa ja virheidenkorjausmenetelmiä jouduttiin korjaamaan monta kertaa testauksen aikana, löydettyjen virheiden takia. Päänvaivaa aiheuttivat virhetilanteet tietokannan luku-tyyppeihin kenttiin viettäessä NULL-arvo. Ongelma ratkaistiin Visual Basicin virheenkäsittelyn On Error Go To -lauseella. Lauseen avulla jatketaan proseduurin suorittamista virheen kohdattaessa.

Tallennus, hakeminen ja poistaminen tietokannasta onnistuivat vaatimusten mukaisesti. Hakulomakkeesta saatiin selkeäkäyttöinen ja kaikki vaaditut eri hakuperusteet saatiin toimimaan.

## Muut mahdolliset toteutusratkaisut

Tarjouslaskuri olisi voitu toteuttaa myös muita ratkaisuvaihtoehtoja käyttämällä. Toteuttamalla ohjelma Visual Basic -kielellä olisi saatu aikaan itsenäinen ohjelma, joka ei olisi riippuvainen isäntäohjelmastaan. Ohjelma olisi saatu käyttämään vähemmän muistia ja ohjelman yleinen ilme olisi voinut olla hieman ammattimaisempi. Tässä vaihtoehdossa ohjelma olisi jouduttu asentamaan jokaiselle koneelle erikseen. Ohjelman päivittäminen ja asetusten sekä muuttuvien tietojen muokkaaminen olisi ollut työläämpää.

Ohjelma olisi voitu myös toteuttaa toimimaan nettiselaimessa esimerkiksi asp-ohjelmointikielen avulla. Graningen asiakkaat olisivat voineet käyttää ohjelmaa suoraan netistä paikasta ja ajasta riippumatta. Tällöin asiakaspalvelun ei tarvitsisi kuluttaa aikaansa niin paljon laskelmien ja vertailujen tekemiseen. Tässä vaihtoehdossa tietoturvasta huolehtimiseen olisi täytynyt käyttää paljon aikaa.

Tarjouslaskuri toteutettiin kuitenkin VBA-kielellä sisäiseksi osaksi Microsoft Exceliä asiakkaan toiveiden mukaan. Vaihtoehto on joustava, koska ohjelma voidaan asettaa suoraan lähiverkkoon, jota mallien avulla voidaan käyttää usean käyttäjän voimin yhtä aikaa. Ohjelman asetuksia ja muuttuvia tietoja on helppoa ja nopeaa muuttaa suoraan Excelin välilehdiltä.

### 4.3 Asiakastestaus ja käyttöönotto

Tarjouslaskuri-ohjelma, tietokanta ja tulosteet asennettiin testiympäristöön Graninge Kainuun lähiverkkoon. Asiakkaalle tehtiin lista, johon asiakas kirjasi löytämänsä virheet, ongelmat, väärin toimivat toiminnot ja mahdolliset kehitystoiveet.

Vaatimusten mukaan ohjelmaa on voitava käyttää usealta työpisteeltä yhtä aikaa. Tarjouslaskurin monen käyttäjän yhtäaikainen käyttäminen mahdollistettiin Excelin mallien avulla. Malli on Microsoft Excel -työkirjapohja, joka tehdään olemassa olevasta työkirjasta, tässä tapauksessa Tarjouslaskurista. Avattaessa malliksi asetettu työkirja, työkirjaan tulee alkuperäisen työkirjan kaikki asetukset ja ominaisuudet käytettäväksi. Mallityökirja avautuu jokaisen käyttäjän paikalliselle koneelle, joten

mallin avulla mahdollistetaan Tarjouslaskurin yhdenaikainen käyttäminen miltä tahansa Graningen lähiverkkoon liitetystä koneelta.

Alkuun asiakastestaukseen osallistui kaksi käyttäjää, mutta parin viikon koekäytön jälkeen testaukseen otettiin lisää käyttäjiä. Asiakastestauksen avulla ohjelmasta löydettiin virheitä. Hintavertailija ja tietokanta selviytyivät asiakastestauksesta ilman suurempia ongelmia, mutta tulosteiden kanssa ilmeni kolme isoa ongelmaa, jotka on esitetty taulukossa 10.

*Taulukko 10. Asiakastestauksen aikana ilmenneet isot ongelmat.*

	Ongelma
1.	Tulosteet aukeavat, tiedot eivät päivity
2.	Tulosteet aukeavat, tiedot eivät päivity seuraavissa Wordin versioissa: XP SP3 ja Word 2003
3.	Tarjouslaskuri kaatuu, jos tulosteita on auki ja avataan uudet tulosteet

Käyttäjän haluamat tulosteet aukesivat, mutta vertailutiedot eivät päivittyneet tulosteisiin. Ongelma johtui Tarjouslaskurin asettamisesta malli-työkirjaksi. Word-tulosteiden viittaukset olivat asetetut lähiverkossa sijaitsevaan alkuperäiseen Tarjouslaskuriin, vaikka viittausten olisi toimiakseen pitänyt viitata jokaisen käyttäjän mallina avattuun työkirjaan. Ongelma korjattiin luomalla proseduuri, joka luo paikallisen koneen C-asemalle Temp-kansion ja sinne TarjouslaskuriTEMP.xls tiedoston, mikäli sitä ei ole jo aikaisemmin luotu. Word asiakirjojen viittaukset asetettiin näyttämään tähän väliaikaiseen tiedostoon ja kaikki hintavertailun tiedot siirretään ohjelmallisesti tähän väliaikaiseen tiedostoon aina ennen tulosteiden avaamista.

Toinen ongelma oli samanlainen ongelma kuin ensimmäinen ongelma, mutta se ilmeni vain kahdessa yrityksen tietokoneessa. Näihin koneisiin oli asennettu Word XP Service Pack 3 ja Word 2003. Ongelman syyksi ilmeni kyseisiin versioihin Microsoftin toimesta parannettu tietoturva. Näissä Wordin versioissa asiakirjaa avattaessa pyydetään vahvistus, saako asiakirjan ulkopuoleisesta lähteestä tuoda tietoja. Ohjelmallisesti yhdistämisen sisältäviä asiakirjoja avattaessa kysymystä ei kysytä ollenkaan, vaan yhdistäminen estetään automaattisesti. Ongelma ratkesi

luomalla Windowsin rekistereihin uuden rekisterin, joka estää kyseisen tietoturva-asetuksen.

Myös kolmas ongelma liittyi tulosteisiin. Käyttäjä ei ole sulkenut avaamiaan tulosteita, suorittaa uuden vertailun ja avaa tulosteet uudelleen, jolloin ohjelma kaatuu. Virheen aiheuttajaksi paikallistettiin asiakastestauksen alussa luotu TarjouslaskuriTEMP.xls tiedosto. Avattaessa tulosteita Tarjouslaskuri siirtää vertailun tiedot tähän väliaikaistiedostoon. Tässä tapauksessa tiedoston tietoja ei voida päivittää, sillä tiedosto on jo käytössä aikaisemmin avattujen tulosteiden viitatessa siihen, joten ohjelma kaatuu ajonaikaiseen virheeseen. Ongelma korjattiin luomalla virheenkäsittely tietojensiirtoproseduriin.

Edellä mainittujen lisäksi ohjelmasta löytyi pieniä virheitä ja ongelmia. Näistä selvittiin pienillä korjauksilla, kuten esimerkiksi virheenkäsittelyn lisäyksillä tai muutoksilla, käyttöoikeuksien antamisella ohjelmiston sijoituspaikkaan ja käyttöliittymän ulkoasun muokkauksilla. Asiakastestauksessa käytetty lista, josta selviää löydetty ongelmat lyhyesti, on esitetty liitteessä D.

Asiakastestauksen aikana huomasin vaiheen tärkeyden ja erityisesti regressiotestauksen merkityksen. Asiakastestauksen avulla ohjelmasta löytyi virheitä, joita ei osattu ottaa huomioon tekijän suorittaman testauksen aikana. Asiakastestauksen kolmas suuri virhe olisi löytynyt heti, mikäli regressiotestaus olisi suoritettu koko tulostus-moduulille ensimmäisen virheen jälkeen.

Tarjouslaskuri otettiin käyttöön 15.3.2005. Käyttöönotto sujui ilman ongelmia ja dramatiikkaa. Suuriosa asiakaspalvelun työntekijöistä oli käyttänyt ohjelmaa jo asiakastestausvaiheessa ja tutustunut sen ominaisuuksiin sekä käyttöön. Työntekijöille pidettiin ohjelman esittely- ja koulutustilaisuus. Käyttöönottoa helpotti myös se, että ohjelman testaus oli onnistunut ja asiakastestauksen aikana oli perehdytty myös ohjelmiston asentamiseen ja konfigurointiin Granningen lähiverkkoon. Ohjelma on toiminut hyvin käyttöönoton jälkeen.



#### 4.4 Jälkihoito

Tarjouslaskuri vaatii tasaisin väliajoin ylläpitoa, jotta esimerkiksi hintatiedot pysyvät ajan tasalla. Energiamarkkinaviraston internetsivuillaan kuukausittain julkaisemat siirto- ja myyntitiedot voidaan päivittää ohjelmaan luodun päivitystoiminnon avulla. Graningen tarjoustuoteryhmät ja niiden eri tariffivaihtoehdot ylläpidetään toistaiseksi käsin niille luoduilla Tarjouslaskurin välilehdillä. Muut muuttuvat tiedot ylläpidetään Ylläpito\_kaavat-välilehdeltä. Päivittäminen ei vaadi ohjelmakoodiin koskemista. Ylläpidosta huolehtimaan koulutetaan joku asiakaspalvelun henkilöistä.

Tarjouslaskuria voidaan kehittää jatkossa. Käyttöönoton jälkeen asiakaspalvelussa on huomattu muutamia uusia ominaisuuksia, joiden avulla ohjelmaa saataisiin kehitettyä parempaan ja monipuolisempaan suuntaan. Mahdollisia muutoksia olisivat muun muassa tietokannan muuttaminen relaatiotietokannaksi, Graningen tarjoustuoteryhmien ja niiden eri tariffivaihtoehtojen päivittämisen helpottaminen sekä automatisoiminen.

#### 4.5 Saavutetut tulokset

Insinööri työn tuloksena saatiin aikaan ohjelma, jonka avulla Graninge Kainuun asiakaspalvelun työskentely nopeutuu ja helpottuu. Tarjouslaskuri-ohjelma syrjäytti vanhan laskuri-ohjelman ja vastasi työn alussa asetettuihin vaatimuksiin. Ohjelma tekee nopeasti tarjouksia sähköstä ja suorittaa hintavertailun eri sähköyhtiöiden välillä. Ohjelman avulla tiedot voidaan tallentaa tietokantaan, joten tietoja on mahdollista käsitellä vertailun suorittamisen jälkeenkin. Tarjouslaskurin avulla voidaan tuottaa nopeasti tarvittavat tulosteet. Liitteessä E on esitetty kuva Tarjouslaskurin 1.0 versiosta.

Käytettävyydeltään Tarjouslaskuri on huomattavasti edistyneempi kuin vanha laskuri, koska kaikki vertailuun liittyvät asiat ovat koko ajan näkyvillä ja valittavissa päälomakkeelta. Ohjelma saatiin toimimaan kaikissa tietokoneissa, jotka jaksavat pyörittää Windows XP:tä ja Office-ohjelmistoja. Tarjouslaskuri toimii ilman ongelmia myös kannettavalla tietokoneella, joka ei ole kytkettynä Graningen lähiverkkoon.

Insinööriyön tuloksena syntyi erilaisia dokumentteja, jotka luovutettiin työn tilaajalle insinööriyön valmistumisen yhteydessä. Näitä dokumentteja ovat vaatimustenmäärittely-, ohjelmisto- ja tämä insinööriyödokumentti. Lisäksi ohjelmaan tullaan tekemään manuaali, jossa selvitetään kuinka ohjelmaa käytetään ja ylläpidetään.

Itselleni insinööriyön tekeminen antoi paljon. Tein ensimmäisen kerran elämässäni suuren ohjelmistokokonaisuuden, jonka aikana käytiin läpi ohjelmistonkehityksen vaiheet määrittelystä testaukseen ja käyttöönottoon. Työn tekeminen opetti monia tärkeitä asioita, kuten ohjelmoimista Visual Basicilla, yleisen ohjelmointiajattelutavan parantumista ja testauksen merkityksellisyydestä ohjelmoinnissa.

## 5 YHTEENVETO

Työn tarkoituksena oli suunnitella ja toteuttaa sähkön tarjoushintojen vertailu-ohjelma, jonka avulla olisi mahdollista tuottaa hintavertailun lisäksi tarjouksia, sopimuksia ja muita asiakirjoja sekä pystyä tallentamaan tehdyt vertailut tietokantaan. Ohjelman tuli korvata Graninge Kainuun vanha ohjelma ja tehdä asiakaspalvelun toiminta helpommaksi sekä nopeammaksi.

Tarjouslaskuri toteutettiin Visual Basic for Applications -ohjelmointikielellä sisäiseksi osaksi Microsoft Excel -sovellusta. Ohjelmaa käytetään graafisen käyttöliittymän avulla. Vertailutiedot tallennetaan Microsoft Access -tietokantaan ja tulosteiden pohjana käytetään Microsoft Word -asiakirjoja.

Tarjouslaskuri täytti työlle asetetut vaatimukset ja se on Graninge Kainuun asiakaspalvelun päivittäisessä käytössä, joten insinööriyttä voidaan pitää onnistuneena. Tarjouslaskuria voidaan kehittää jatkossa uusilla ominaisuuksilla ja toiminnoilla, joiden avulla ohjelmasta saadaan vieläkin monipuolisempi.

Insinööriyden tekeminen tuntui aluksi todella vaikealta tehtävältä. Toisaalta työn laajuus ja haastavuus toimivat motivaation lähteenä työtä tehdessä. En ollut ohjelmoinut Visual Basicilla aikaisemmin paljoakaan ja tietokantojen suunnittelusta sekä toteutuksesta ei ollut aikaisempaa kokemusta. Työn ensimmäinen kuukausi vierähti kokonaan Visual Basic -kielen, tietokantojen ja hintavertailujen opiskeluun. Kokonaisuudessaan insinööriyden suorittamiseen kului lähes puolivuotta aikaa.

Tarjouslaskurin suunnittelemisen ja toteuttaminen oli erittäin antoisa ja opettavainen kokemus. Ohjelmiston kehityksen vaiheet ja niihin liittyvät asiat käytiin läpi käytännössä, yleisestä ohjelmointiajattelusta sai paljon oppia ja Visual Basic -kieli tuli tutuksi. Työn aikana esiintyneistä ongelmista selvittiin kovalla tutkimustyöllä ja erilaisia toteutusratkaisuja kokeilemalla. Työn tilaajan kanssa käydyistä keskusteluista ja katselmoinneista sai korvaamatonta apua työn valmistumisen kannalta. Testauksen tärkeyden merkitys tuli opittua, ajoittain jopa kantapään kautta.

## LÄHDELUETTELO

- 1           Energiamarkkinavirasto. Sähkön hintavertailu. 2002.  
Päivitetty 17.2.2005. [WWW-dokumentti].  
<<http://www.energiamarkkinavirasto.fi/calc1.asp>>.
- 2           Haikala, I. & Märijärvi, J. Ohjelmistotuotanto. Kahdeksas uudistettu  
painos. Pieksämäki: RT-Print, 2002. ISBN 952-14-0486-8
- 3           Scott, D.F. MS Office 2000 -sovelluskehitys. Ensimmäinen painos.  
Jyväskylä: Teknolit Oy, 2000. 5 s. ISBN 951-846-015-9
- 4           VBA vs Visual Basic vs VBScript. [WWW-dokumentti].  
<<http://www.natecsystems.com/98%20Sep-Oct%20VBA%20vs%20Visual%20Basic.html>>.  
(Luettu 27.2.2005.)
- 5           Jacobson, R. Excel 2000 ohjelmointi; Visual Basic for Applications.  
Helsinki: Oy Edita Ab, 2000. 6-7 s. ISBN 951-826-069-9
- 6           Halvorson, M. & Young, M. Running Office 97. Espoo: Suomen Atk-  
kustannus Oy. 1997. 833-834 s. ISBN 951-762-526-X
- 7           Ek, J. , Eriksson, U. & Isanovic, S. VBA-ohjelmointi Office 2000:ssa.  
Vantaa: Tummavuoren kirjapaino Oy, 2000. 238 s. ISBN 951-05-  
1109-4
- 8           Rahmel, D. Visual Basic 6: Tietokantaohjelmointi. Helsinki: IT Press,  
1999. ISBN 951-826-078-8.
- 9           Halvorson M. Microsoft Visual Basic .NET Trainer Kit. Helsinki: IT  
Press, 2002. 352 s. ISBN 951-826-653-0
- 10          Hovi, A. Huotari, J. & Lahdenmäki, T. Tietokantojen suunnittelu &  
indeksointi. Ensimmäinen painos. Jyväskylä: Docendo Finland Oy,  
2003. ISBN 951-846-178-3.
- 11          Lahtonen, T. SQL. Toinen painos. Jyväskylä: Docendo Finland Oy,  
2002. ISBN 951-846-092-2.
- 12          Eriksson, H. & Penker, M. UML. Helsinki: IT Press, 2000.  
ISBN 951-826-026-5

Kentän nimi	Tietotyyppi	Arvo tarvitaan	Oletusarvo	Rajoitus
VertailuID	Laskuri	Kyllä	-	Uniikki
Sukunimi	Teksti	Ei	Ei	30 merkkiä
Etunimi	Teksti	Ei	Ei	20 merkkiä
Osoite	Teksti	Ei	Ei	40 merkkiä
Pnumero	Luku	Ei	Ei	Ei
Ppaikka	Teksti	Ei	Ei	30 merkkiä
PuhNro	Teksti	Ei	Ei	30 merkkiä
Email	Teksti	Ei	Ei	40 merkkiä
OsoiteKP	Teksti	Ei	Ei	40 merkkiä
PnumeroKP	Luku	Ei	Ei	Ei
PpaikkaKP	Teksti	Ei	Ei	30 merkkiä
KayttoNroKP	Teksti	Ei	Ei	10 merkkiä
Sulakekoko	Teksti	Kyllä	Ei	10 merkkiä
KulutusKok	Luku	Kyllä	Ei	Ei
KulutusKallis	Luku	Kyllä	Ei	Ei
KulutusHalpa	Luku	Kyllä	Ei	Ei
KulutusKPros	Luku	Kyllä	Ei	Ei
KulutusHPros	Luku	Kyllä	Ei	Ei
Siirtoyhtio	Teksti	Kyllä	Ei	50 merkkiä
Siirtotuote	Teksti	Kyllä	Ei	50 merkkiä
SPerusmaksu	Luku	Ei	Ei	Ei
SEmaksu1	Luku	Kyllä	Ei	Ei
SEmaksu2	Luku	Ei	Ei	Ei
SEmaksu3	Luku	Ei	Ei	Ei
SEmaksu4	Luku	Ei	Ei	Ei
SHintaYht	Luku	Kyllä	Ei	Ei
Myyntiyhtio	Teksti	Kyllä	Ei	50 merkkiä
Myyntituote	Teksti	Kyllä	Ei	50 merkkiä
MPerusmaksu	Luku	Ei	Ei	Ei
MEmaksu1	Luku	Kyllä	Ei	Ei
MEmaksu2	Luku	Ei	Ei	Ei
MEmaksu3	Luku	Ei	Ei	Ei
MEmaksu4	Luku	Ei	Ei	Ei
MHintaYht	Luku	Kyllä	Ei	Ei
KokHinta1	Luku	Kyllä	Ei	Ei
GGKtarjous	Teksti	Kyllä	Ei	50 merkkiä
GGKtuote	Teksti	Kyllä	Ei	50 merkkiä
GGKperusmaksu	Luku	Ei	Ei	Ei
GGKEmaksu1	Luku	Kyllä	Ei	Ei
GGKEmaksu2	Luku	Ei	Ei	Ei
GGKEmaksu3	Luku	Ei	Ei	Ei
GGKEmaksu4	Luku	Ei	Ei	Ei
GGKHinta	Luku	Kyllä	Ei	Ei
KokHinta2	Luku	Kyllä	Ei	Ei
EroVuodessa	Luku	Kyllä	Ei	Ei
Aloituspvm	Pvm./klo	Kyllä	Ei	Lyhyt pvm.
Lopetuspvm	Pvm./klo	Kyllä	Ei	Lyhyt pvm.
Tekijä	Teksti	Kyllä	Ei	50 merkkiä
Vertailupvm	Pvm./klo	Kyllä	Ei	Lyhyt pvm.

Matti Meikäläinen  
Testie 10  
00100 Helsinki

Puhelinnumero	044-1231234
Käyttöpaikan osoite	Testie 10 00100 Helsinki
Pääsulake	3*25 A
Sähkönkäyttö	3613 kWh/vuosi
päivä tai talviarkipäivät	1228 kWh 34 %
yö tai muu aika	2385 kWh 66 %

## Sähkön myynnin hintavertailu

### Sähkön myynti

Vertailtavina ovat tarjoushintamme ja Vantaan Energia Oy:n hinnaston mukainen hinta tai asiakkaan ilmoittama hinta.

Myyntiyhtiö			Graininge Kainuu Oy	Vantaan Energia Oy
Sopimustuote	Sähkö 2			
Myyntituote	Yösähkö 1			Aikasähkö
	Sopimushinnat 23.3.2005			Myyntihinnat 1.1.2004
Perusmaksu, euroa/kk	3,00			3,50
Energia, snt/kWh	päivä tai talviarkipäivät	4,15		4,70
	yö tai muu aika	3,15		3,10
Yhteensä euroa/vuosi	162,09			173,65
ETU euroa/vuosi	11,56			

### Sähkön siirto

Sähkön siirtomaksut veloittaa verkkoyhtiö kulloinkin voimassaolevan hinnastonsa mukaan.

Verkkoyhtiö			Helsingin Energia
Siirtotuote			Aikasiirto
Kaksiaikaisissa sähkötuotteissa aikajaksotus noudattaa verkkoyhtiön käytäntöä.			Siirtohinnat 1.1.2004
Perusmaksu, euroa/kk			8,54
Energia, snt/kWh	päivä tai talviarkipäivät	1.1-31.12 ma-pe 7-20	3,32
	yö tai muu aika	1.1-31.12 ma-pe 20-7, la-su 0-24	1,67
Sähkövero, snt/kWh			0,91
Yhteensä euroa/vuosi			215,96

### Sähkön kokonaishinta

= sähkön myynti + sähkön siirto sähköveroineen yhteensä

Myyntiyhtiö	Graininge Kainuu Oy	162,09	Vantaan Energia Oy	173,65
Verkkoyhtiö	Helsingin Energia	215,96	Helsingin Energia	215,96
Yhteensä euroa/vuosi		378,05		389,61
ETU euroa/vuosi		11,56		

Hinnat sisältävät arvonlisäveroa 22 %.  
Hintalähde: Energiamarkkinavirasto (www.energiamarkkinavirasto.fi)

# Sähkönmyyntisopimus

Tällä sopimuksella Graninge Kainuu Oy sähkönmyyjänä (jäljempänä Myyjä) ja sähkönostaja (jäljempänä Ostaja) sopivat sähkökaupasta seuraavaa:

<b>Myyjä</b>	Graninge Kainuu Oy, PL 5, 87101 Kajaani, puh. (08) 61511, kotipaikka Kajaani, Y-tunnus 0185353-1		
<b>Ostaja</b>	Nimi	<b>Matti Meikäläinen</b>	Henkilötunnus <sup>*)</sup>
	Laskutusosoite	<b>Testie 10</b>	<sup>*)</sup> pakollinen tieto, ks. sopimusehtojen 1 §
	Postitoimipaikka	<b>00100 Helsinki</b>	
	Puhelinnumero	<b>044-1231234</b>	
	Sähköpostiosoite	<b>matti.meikalainen@hesa.fi</b>	
	Sähkönkäyttöpaikan osoite ja postitoimipaikka		
		<b>Testie 10, 00100 Helsinki</b>	
	Käyttöpaikan sulakekoko	<b>3*25 A</b>	Käyttöpaikan numero <b>1234567</b>
	Verkkoyhtiö	<b>Helsingin Energia</b>	
	Nykyinen myyjä	<b>Vantaan Energia Oy</b>	
	Ryhmitunnus (Myyjä täyttää)		

<b>Sopimustuote</b>	<b>Sähkö 2</b>
<b>Myyntituote</b>	<b>Yösähkö 1</b>
<b>Myyntihinta</b>	Perusmaksu, euroa/kk 3,00
<b>sopimusaikana</b>	Energia, snt/kWh 4,15 päivä tai talviarkipäivät
	3,15 yö tai muu aika

Hinnat sisältävät arvonlisäveroa 22 %.

<b>Sopimuskausi</b>	Sopimus on määräaikainen. Sopimuskausi alkaa a) 3 viikon kuluttua sopimuksen palautumisesta Myyjälle tai b) jos Ostajalla on voimassaoleva määräaikainen sopimus, / 2005 alkaen edellyttäen, että sopimus on palautunut Myyjälle vähintään 2 viikkoa ennen em. päivämäärää. Sopimuskausi päättyy <b>31.3.2007</b> , minkä jälkeen sopimus jatkuu sopimusehtojen mukaisesti toistaiseksi voimassaolevana.
---------------------	--

## Laskutusvalinnat

<b>Määrä</b>	Laskuja vuodessa <input type="checkbox"/> 2 <input type="checkbox"/> 4 <input type="checkbox"/> 6 tai <input type="checkbox"/> 12 (valittavissa, jos sähkönkäyttö yli 10 000 kWh/vuosi)
<b>Eräpäivä</b>	Maksukuukausien _____ . päivä
<b>Lähetystapa</b>	<input type="checkbox"/> Laskut sähköpostitse (ilmoita sähköpostiosoite yllä)
<b>Suoraveloitus</b>	Tulee voimaan, jos oheinen suoraveloitusvaltakirja toimitetaan pankkiin.
<b>Siirtotuote</b>	Sähkön siirtotuote säilyy ennallaan. Verkkoyhtiö veloittaa sähkön siirtomaksut ja sähköveron kulloinkin voimassaolevan hinnastonsa mukaan.
<b>Tarjous voimassa</b>	Sopimustarjous on voimassa <b>13.4.2005</b> saakka.

Paikka ja päiväys

Paikka ja päiväys

**Kajaani**

**23.3.2005**

Ostajan allekirjoitus

**Graninge Kainuu Oy**

Asiakaspalvelu

**Matti Meikäläinen**

**Aki Martikainen**

Kuvaile virhe	Nimi	Päiväys	Korjaustiedot
Tulosteiden vertailutiedot eivät päivity ollenkaan	Aki	17.2.2005	ok. Korjattu 17.2. / amar
Tulosteiden vertailutiedot eivät päivity Word XP SP3 ja Word 2003:ssa	Aki	18.2.2005	ok. Korjattu 21.2. / amar
Myyntituote ei tulostu myyntisopimukselle	Jaana	18.2.2005	ok. Korjattu 18.2. / amar
Allekirjoitusten korjaus muotoon Etunimi Sukunimi, sopimukset	Jaana	18.2.2005	ok. Korjattu 18.2. / amar
Syötin asiakkaan tiedot + päivä- ja yökulutuksen: virheilmoitus: compile error: can't find project or library	Jaana	21.2.2005	ok. Korjattu 21.2. / jpen
Vertailussa näkyy myynnin osalla kellonajat, mutta ei tulostuksessa	Seija	21.2.2005	ok.
Sopimuksella word-pohja herjaa "reunukset ulkopuolella, haluatko jatkaa k/e)	Seija	21.2.2005	ok. Tulosteiden ylläpitäjä korjaa
Tallennus tietokantaan: tallentaako tuplasti samat tiedot jos painaa vahingossa tallenna-nappia uudelleen	Jaana	24.2.2005	Ominaisuus, toistaiseksi näin amar / jpen
Ei anna kirjoittaa oletustekstiä siirto- ja myyntiyhtiönimeen: gran... ja tulisi yhtiön nimi	Seija	24.2.2005	Käyttäjävirhe, toimii
Tallennusvaiheessa meni jumiin: olin valinnut molemmat täpät; sopimus ja vertailu: run-time error 1004	Seija	24.2.2005	Sulje wordin avoinna olevat tiedostot
Laskee säästövuodenaikeasähkön julkiset hinnat väärin ja ajat ristissä ks. hintavertailua -lomakkeella	Jaana	24.2.2005	Ylläpito korjannut, jket ilmoitti thaa:lle 28.2.
Vattenfallin siirron perusmaksut kerros- ja rivitaloissa ei anna perusmaksua ollenkaan. EM:n sivujen korjaus?	Seija	3.3.2005	Virhe EMV:n julkaisemissa listoissa, ylläpito korjaa
Tein vertailua asiakkaalle, oma alue, ensin säästöyösähköllä, sitten säästövuodenaikeasähköllä. Herjasi tallennus- tai tulostusvaiheessa "tässä sijainnissa on jo tiedosto TarjouslaskuriTEMP.xls. Haluatko korvata sen? Vastasin En. Aloin tulostaa, tuli herja Run time error 1004. TarjouslaskuriTEMP.xls on kirjoitussuojattu.	Jaana	3.3.2005	Sulje wordin avoinna olevat tiedostot
Asiakkaan nimi: avopari, jolla omat etu- ja sukunimet. Tallensin molemmat sukunimen kohdalle: herjasi: virhe tk tallennuksessa. Virhe - 2147217887: Lisättävät tiedot ei mahdu kenttään. Yritä lisätä / vähentää tietoja. Asiakas Sari Vornanen ja Marko Mansikkala. Korjasin niin, että Marko etunimen kohdalle.	Jaana	3.3.2005	ok. Tietokannan kentän kokoa muutettu 4.3. / amar
Tulosteiden avaamisen yhdydessä syntyy ajonaikainen virhe, jos tulosteita ennestään auki (runtime error 1004)	Aki	14.3.2005	ok. Korjattu 14.3. / amar
0 -alkava postinumero ei tulostu sopimukselle kokonaan: ensimmäinen/ensimmäiset nollat jää pois. esim. Väliniemi Minnan sopimus	Slie	16.3.2005	ok. Korjattu 22.3. / amar
pdf -muodossa kun tallennetaan sähkönymyyntisopimuskoitalous, tulee allekirjoitusten selvennykset seuraavalle sivulle	Slie	16.3.2005	jpen/jket muotoilevat tulosteita uudelleen
halpahalli -yrittäjähinta 2-vuoden sopimukseen tulee laskurille energian osalta eri kuin ylläpito hinnasto. Ylläpito hinnastossa 3,79, laskurissa 3,65 (alv 22 %) esim. Aho Tuomo (2 tallennusta), toiseen muutin riville oikeat hinnat. Lähetetty asiakkaalle alvton tarjous.	slie	16.3.2005	Hinnaston ylläpitäjä korjaa
tulostin ekan spimuksen ja vertailun+saatteen testiohjelmasta. Varsinaisessa sopimuslomakkeessa tekstit menevät liikaa oikealle	Eeva	16.3.2005	täsmennätkö Eeva tätä



**Tarjouslaskuri - Versio 1.0**

**Asiakastiedot**

Sukunimi: Meikäläinen

Etunimi: Matti

Katuosoite: Testie 10

Postinumero: 00100

Postitoimipaikka: Helsinki

Puhelinnumero: 044-1231234

Sähköpostiosoite: matti.meikalainen@hesa.fi

**Käyttöpaikan tiedot**

Katuosoite: Testie 10

Postinumero: 00100

Postitoimipaikka: Helsinki

Käyttöpaikanro: 1234567

**Asiakas- ja käyttöpaikkatiedot**

Hae vertailu Tyhjennä

**Sähkönkäyttötiedot**

Yhteensä kWh/v: 3613 Apulinkki

Pv/talviarkipv: 1228 34 %

Yö/muu aika: 2385 66 %

Tarkista käyttötiedot Tarkista luvut

Sulakekoko: 3\*25 A

Tyhjennä / vaihda tuote

**Vertailutiedot: Yö-/vuodenaikäsähkö**

Siirtoyhtiö: Helsingin Energia Myyntiyhtiö: Vantaan Energia Oy

Siirtotuote: Myyntituote

Tuote	e/kk	snt/kWh	Voimassaoloaika
Aikasiirto	8,54	3,32	1.1-31.12 ma-pe 7-20
Ohjattu yösiirto	8,54	1,67	1.1-31.12 ma-pe 20-7, la-su 0-24

**GGK vertailutiedot: Yö-/vuodenaikäsähkö**

Tarjoustuote: Muu Suomi, kaksi vuotta

Sopimus alkaa: 13.4.2005 Sopimus päättyy: 31.3.2007

Tuote	e/kk	snt/kWh	Voimassaoloaika
Yösähkö 1	3,00	4,15	1.1-31.12 ma-su 7-22
Yösähkö 2	3,00	3,15	1.1-31.12 ma-su 22-7
Vuodenaika 1	3,00		
Vuodenaika 2	3,00		

Yösähkö 1 3,00 4,15 3,15

**Myyntiyhtiöt**

Yhtiön nimi: Kunta

**Vertailupainikkeet**

Suorita vertailu Katso vertailua

**Hintavertailu**

e/vuosi e/vuosi

Graniinge Kainuu 378,05 Nykyinen yhtiö 389,61

Erotus vuodessa 11,56 **Hinnat + alv**

**Valitse tulostettavat asiakirjat**

☐ Sähkönhintavertailu

☒ Sähkönmyyntisopimus kotitalous

☐ Sähkönmyyntisopimus yritys

**Tämän tarjouksen teki**

Aki Martikainen

**Toimintapainikkeet**

Avaa valitut Tallenna Poistu laskurista

© Graniinge Kainuu Oy